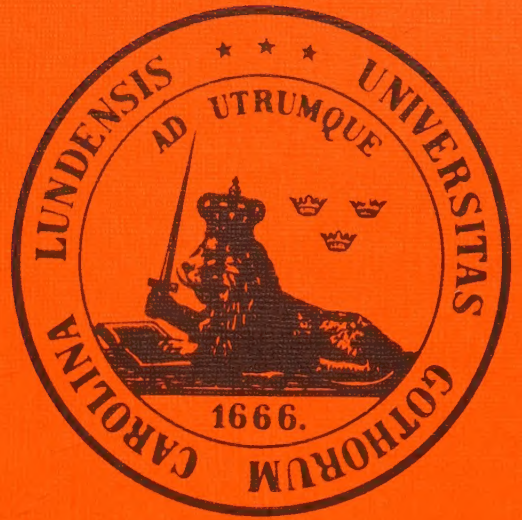




INTERAKTIV GENERERING AV GEOMETRI MED GRAFISKT OCH DIALOGT DATORSTÖD VID PROGRAMMERING AV NC-SVARVAR

Sven Åke Eriksson

**Institutionen för mekanisk teknologi och verktygsmaskiner
Lunds Tekniska Högskola
1979**



INTERAKTIV GENERERING AV GEOMETRI MED GRAFISKT OCH DIALOGT DATORSTÖD VID PROGRAMMERING AV NC-SVARVAR

Av

SVEN ÅKE ERIKSSON

Civ ing Sydsåkanska nationen.

AKADEMISK AVHANDLING

som för avläggande av teknisk doktorsexamen vid
tekniska fakulteten vid universitetet i Lund kommer
att offentligen försvaras å hörsal M:B, Maskin-
teknik, Ole Römers v. 1, Lund, lördagen den
26 maj 1979 kl 9.

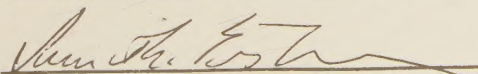
Organisation LUND UNIVERSITY Department of Production and Materials Engineering Box 725 S-220 07 LUND 7		Document name DOCTORAL DISSERTATION	
		Date of issue May 1979	
Author(s) Sven Åke Eriksson		CODEN: LUTMDN/(TMMN-1001)/001-64/(1979)	
		Sponsoring organisation	
Title and subtitle Interaktiv generering av geometri med grafiskt och dialogt datorstöd vid program- mering av NC-svarvar - Interactive generation of geometry for NC-lathes with graphical and dialogue computer aid.			
Abstract			
The <u>technological</u> and <u>economical</u> environment for part programming of <u>numerically controlled machine tools</u> is analysed and described for customer oriented <u>small lot production</u>. To meet the demands from this type of production an easy-to-learn method for <u>generation of part geometry</u> is specified. The method for description of geometry uses an <u>interactive graphics minicomputer system</u> to provide extensive <u>computer aid</u> including <u>menues</u> and <u>graphics</u>. To facilitate for NC-lathe operators to use the system for part programming the existing software is modified and enlarged with a dialogue which guides the operator through the programming work. In this dialogue the geometry is described with <u>standardised elements</u> which are familiar to the operator. The element descriptions are built to make further <u>automation</u> of the part programming possible by adding and retrieving <u>not geometrical information</u> to each element. The system will provide increased flexibility to the part programming of numerically controlled machine tools and give possibilities for <u>job enrichment</u>.			
Key words			
Classification system and/or index terms (if any)			
Supplementary bibliographical information			Language Swedish
ISSN and key title			ISBN
Recipient's notes		Number of pages 64	Price
		Security classification	

Distribution by (name and address)

Lund University, Dept of Production and Materials Engineering

I, the undersigned, being the copyright owner of the abstract of the above-mentioned dissertation, hereby grant to all reference sources permission to publish and disseminate the abstract of the above-mentioned dissertation.

Signature



Date

May 1979



Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41553413_0030

INTERAKTIV GENERERING AV GEOMETRI MED GRAFISKT OCH DIALOGT DATORSTÖD VID PROGRAMMERING AV NC-SVARVAR

Sven Åke Eriksson





FÖRORD

Föreliggande arbete behandlar en ny systemfilosofi för automatisk programmering av numeriskt styrda verktygsmaskiner, och beskriver en praktisk lösning av delproblemet numerisk geometribeskrivning vid svarvprogrammering.

Den kritiske läsaren som frågar sig varför ännu en systemfilosofi för automatisk programmering presenteras, när det redan existerar ett stort antal färdiga system på marknaden, kommer att finna att de här presenterade tankegångarna tillgodoser helt nya krav. Det är krav som ställs på grund av nya arbetsformer och begäran om medinflytande som växer sig allt starkare i den mekaniska verkstadsindustrin. Dessa krav måste utnyttjas för att ekonomiskt kunna förverkliga stegen mot ökad automation av tungt och påfrestande arbete.

Projektet har genomförts på initiativ från maskinverkstaden vid Kockums Varv AB i samarbete med institutionen för mekanisk teknologi och verktygsmaskiner vid Tekniska Högskolan i Lund och med finansiellt stöd av Styrelsen för Teknisk Utveckling. En mycket stor del av arbetet har utförts på den interaktiva grafiska datoranläggning som installerats vid institutionen för mekanisk teknologi.

Genom att projektet utformats som ett samarbetsprojekt med tekniska högskolan har jag och min kollega där, civ ing Dan Johannesson, givits utomordentliga möjligheter att behandla våra respektive delar av programmeringsproblematiken förutsättningslöst. Efter gemensamma inledande diskussioner har vi delat upp svarvprogrammeringen i ett arbete som beskriver den numeriska definitionen av arbetsstyckets geometri och ett annat arbete som beskriver generering av tillhörande verktygsvägar för att åstadkomma övergången från råämne till färdigvarvad detalj. Nedanstående avhandling beskriver genereringen av den numeriska geometriinformation medans Dan Johannessons arbete "Generering av verktygsbanor för NC-svarvar på ett interaktivt grafiskt minidatorsystem" beskriver genereringen av verktygsvägar. Tillsammans avser avhandlingarna ge ett förslag till hur man kan lösa programmeringsarbetet av verktygsmaskiner med utnyttjande av interaktiva grafiska datorsystem.

Under arbetets gång har ett stort antal människor bistått mig med råd och förslag. Främst vill jag tacka chefen för den produktions-tekniska förvaltningen vid Kockums Varv AB dir Paul Pålsson som har verkat för att interaktiv teknik skall användas för programmeringsarbetet, planeringschefen vid maskinverkstaden Jan Bergqvist för stöd och uppmuntran av projektets utformning samt min handledare professor Gunnar Kullberg som genom stort personligt engagemang gjorde det möjligt för mig och min kollega Dan Johannesson att få arbeta med den nya typen av datorsystem.

INNEHÅLLSFÖRTECKNING

Sid

1	BAKGRUND	
1.1	Maskinverkstaden	1
1.2	NC-teknik i maskinverkstaden	1
1.3	Tillverkningssortiment	2
1.4	Problemformulering	3
2	SYFTE OCH AVGRÄNSNINGAR	
2.1	Syfte	4
2.2	Avgränsningar	4
3	METODBESKRIVNING	
3.1	Kronologiskt schema	5
3.2	Analys av detaljkomplexitet och bearbetningsteknik	6
3.3	Jämförande analyser av tillverkningskostnaden vid konventionell bearbetning och numeriskt styrd bearbetning	7
3.4	Kartläggning av befintlig NC-organisation	9
4	SYSTEMMILJÖ	
4.1	Produktsortiment	11
4.2	Tillgänglig verktygsmaskinutrustning	13
4.3	NC-verksamhetens organisation	15
5	SYSTEMMODELL	
5.1	Arbetsplatsutformningen	17
5.2	Dialogutformning	25
5.3	Funktionsutformning	28
6	PROGRAMTEKNISKT UTFÖRANDE	
6.1	PEP-version	33
6.2	Fortranversion	34
6.3	Grafisk bildhantering	36

	Sid
7 SYSTEMREALISERING OCH VIDAREUTVECKLING	38

SUMMARY IN ENGLISH

FIGURBILAGA

LITTERATURFÖRTECKNING

1 BAKGRUND

Avhandlingen beskriver tankegångarna bakom ett datorbaserat programmeringssystem som fortsättningsvis kallas IPS. Denna förkortning står för Integrerat Produktions System som kan härledas från generella diskussioner i samband med Kockums Maskinverkstads satsning på CNC-teknik.

Eftersom IPS-systemet är tänkt att kunna användas i maskinverkstaden vid KOCKUMS VARV AB, ges nedan en summarisk beskrivning av den aktuella delen av varvet.

1.1 MASKINVERKSTADEN

Organisatoriskt är maskinverkstaden en avdelning inom produktionsförvaltningen vid KOCKUMS VARV AB. Storleksmässigt kan den liknas vid en medelstor verkstadsindustri med ca 500 anställda.

Kopplingen till varvet har medfört att maskinverkstaden är dimensionerad för att kunna hantera och bearbeta arbetsstycken med mycket stora dimensioner (se fig 1), såväl som mera normal produktion.

För att kunna möta alla de olika krav fartygsproduktionen vid varvet ställer, har man praktiskt taget alla typer och storlekar av maskiner till sitt förfogande. De för detta arbete intressanta maskinerna är de numeriskt styrda (NC och CNC). Det finns fyra åtta stycken numeriskt styrda maskiner, varav två NC-styrda arborrverk, två CNC-styrda karusellsvarvar och en CNC-styrd support-svarv med automatisk verktygsväxling och vardera en koordinatborrmaskin, ett arborrverk och en axelsvarv, samtliga CNC-styrda.

1.2 NC-TEKNIK I MASKINVERKSTADEN

När projektet initierades i januari 1974 fanns det enbart två NC-styrda arborrverk i maskinverkstaden. Dessa är utrustade med tidigare generationers styrsystem som inte medger någon mera avancerad produktionsteknisk arbetsmetodik. Beslut om utökning av NC-kapaciteten med en karusellsvarv, en axelsvarv, en support-svarv, en koordinatborrmaskin och ett arborrverk, alla med CNC-styrningar, hade fattats och maskinerna levererades och togs i drift under 1974 och 1975.

Programmeringsfunktionen

När CNC-satsningen beslutats skickades ett 40-tal man av maskinverkstadens personal på grundläggande utbildning och information om hur tekniken fungerar. Denna breda information om CNC-tekniken syftade till att förhindra att alltför höga murar skulle växa upp mellan den konventionella delen av verkstaden och de numeriskt styrda maskinerna.

Den redan tidigare befintliga programmeringsfunktionen bestod av tre man, som på grund av det kraftigt ökade programmeringsbehovet utökades med fyra man. Dessa fyra man togs ut bland de kollektivanställda maskinoperatörer som visat intresse för de nya arbetsmoment som tillkommit vid CNC-maskinerna.

Programmeringshjälpmedel

Den grundläggande utbildningen avsåg s k manuell programmering av maskinerna. Denna metod har nästan uteslutande använts för att taga fram de ca 1200 program som finns i programbiblioteket.

Långt innan CNC-satsningen i maskinverkstaden, hade Kockums data-avdelning utvecklat ett geometrihanterande dataspråk, Styrbjörn, som användes för skrovproduktionen inom företaget. En utvidgning av detta system med specialutvecklade rutiner för verktygsmaskinerna avsåg att utgöra grunden för ett s k automatiskt programmeringssystem.

På grund av att Styrbjörnsystemet utnyttjar företagets centrala dator och denna inte medger omfattande terminalkörningar för maskinverkstadens ändamål, har systemtankegångarna inte fullföljts.

Samtidigt med svårigheterna att uppnå ett centraldatorbaserat system som uppfyllde maskinverkstadens krav visade ett samarbete med institutionen för mekanisk teknologi och verktygsmaskiner vid LTH på möjligheten att utnyttja fristående minidatorbaserade system.

1.3 TILLVERKNINGSSORTIMENT

När CNC-satsningen beslutades 1973 var maskinverkstadens produktion inriktad på att förse varvet med komponenter till seriebyggda tankers och gasfartyg. Framdrivningsmaskineriet i dessa gav en relativt stor andel serietillverkade turbindetaljer. Denna serieproduktion av tankers har under de senaste åren ersatts av styckeproducerade fartyg med därtill hörande engångstillverkade detaljer.

Dessutom har det ökade antalet alternativa produkter medfört en inriktning mot allt kortare tillverkningsserier. Dessa produkter är så olika sinsemellan att man endast i ringa omfattning kan utnyttja tidigare erfarenheter, utan i stort sett måste betrakta varje detalj som helt ny.

Förutom denna produktion som är direkt beroende av varvets nybyggnadsprogram har maskinverkstaden en kontinuerlig grundbeläggning av dieselmotorreservdelar. Eftersom det är frågan om förslitningsdetaljer som ex vis kolvar och cylinderfoder är behovet och därför även produktionen stabil.

Då man har tillgång till smedja, gjuteri och maskinbearbetning inom samma produktionsavsnitt har maskinverkstaden alltid till viss del belagts med legoproduktion.

Det ständigt växlande produktionssortimentet som bearbetas i maskinverkstaden ställer därför krav på enkel, snabb och säker tillverkningsunderlagsframställning.

1.4 PROBLEMFORMULERING

Under 1973 diskuterades en förnyelse av vissa delar av maskinparken i maskinverkstaden för att möta de nya fartygens högre krav på precision och för att ge verkstaden en ökad konkurrensförmåga inom skärande bearbetning. Man förde då djupgående diskussioner med andra stora användare av NC-teknik. Därvid bekräftades vikten av att man ägnar stort intresse åt programmeringsfrågor, produktionsförberedelsearbete samt bemannings- och beläggningsfrågor vid NC-bearbetning.

NC-teknik ställer andra krav på maskinoperatörer, förmän och beredare än konventionell teknik. Eftersom yrkeskunnandet sitter i hållremsan vid NC-bearbetning anser vi det orimligt att låta skickliga maskinoperatörer stå vid sidan om den automatiskt arbetande maskinen bara för att byta arbetsstycke eller för att hålla fritt från spånor. Samtidigt med avlastningen av operatörens arbetsuppgifter ger NC-tekniken en kraftig ökning av de produktionsförberedande uppgifterna genom att den kräver noggrannare planering och en detaljerad beskrivning av hela bearbetningsförloppet. En del av maskinoperatörens arbete har således flyttats bakåt i informationskedjan och utföres oftast hos beredarna på verkstadskontoret. I dessa arbetsuppgifter ingår också förinställning av verktyg som en ny funktion.

Det är tydligtvis en betydande omfördelning av arbete som ägt rum. Genom att de arbetsledande funktionerna oftast inte heller till alla delar förstår den nya tekniken är det lätt hänt att irritationen växer om maskiner står stilla p g a fel i program eller andra fel utanför verkstadens kontroll.

Problemet är av dessa skäl mycket komplext och kommer att ytterligare förstärkas allt eftersom kraven ökar på två- eller flerskifts utnyttjande av kapitalintensiv utrustning. Vi kan konstatera att vi är på väg mot en utveckling som genom hittills gjorda rationaliseringsåtgärder haft följande effekt.

Genom uppdelningen av maskinoperatörens arbetsuppgifter på programmerare, verktygsförinställning och maskinbetjäning har vi fått ett delat och opersonligt ansvar för produktionen i maskinerna. Den komplicerade tekniken att beskriva vad verktygsmaskinen skall utföra har vidare lett till försvårad kommunikation mellan berörd personal och en brist på programmeringskunnig personal medan den yrkesskickliga personalen saknar möjlighet att utföra programmeringsarbetet.

Mot denna bakgrund startade vi ett projekt som skulle analysera problemområdet och ange vägar och teknik för att bättre utnyttja befintliga resurser.

2 SYFTE OCH AVGRÄNSNINGAR

2.1 SYFTE

Under ett antal studiebesök i början av projektet såg vi hur man hade löst programmeringsfrågan på olika industrier. Vi kunde då konstatera att det fanns ytterligare ett antal viktiga parametrar som kunde påverka beslutet om vilken programmeringsteknik som skulle användas för maskinverkstadens NC-maskiner.

För att skaffa en så god bild som möjligt över NC-bearbetningens totalekonomi bestämdes det att projektets ena huvudsyfte var att undersöka och belysa alla påverkande faktorer. Med alla fakta tillgängliga skulle därefter en kravspecifikation över ett idealt system upprättas.

När vi nått dit visade det sig lockande att utnyttja interaktiva grafiska datorsystem för att åstadkomma en metodorienterad datorstödd lösning på programmeringsarbetet. Tidigare hade sådana system främst använts för konstruktionsändamål och nästan enbart hos de stora flygplanstillverkarna i USA. Eftersom systemen just hade börjat säljas i Europa beslutade vi oss för att undersöka om och hur man skulle kunna använda tekniken för att taga fram ett programmeringssystem som svarade mot kravspecifikationen.

2.2 AVGRÄNSNINGAR

Med utgångspunkt från tillverkningen i maskinverkstaden begränsades projektet till att i första hand gälla svarvningsoperationen.

Den främsta orsaken till denna begränsning är att svarvning svarar för större delen av NC-bearbetningen i verkstaden. Genom att arbetsoperationerna vid svarvning dessutom är korta leder det till ett mycket stort behov av nya program. Man tvingas då välja detaljer som tillverkas i långa serier för att få andrum och hinna med nyprogrammering och uttestning. Denna dragning mot långserieproduktion är emellertid inte önskvärd eftersom man då inte utnyttjar NC-maskinens fördelar av snabb omställning för olika arbetsobjekt.

Genom att välja svarvningsoperationen räknar vi med att det för-
enklade programmeringsförfarandet kommer att ge snabba och stora effekter på användningen av NC-teknik för kortserietillverkning. Vi anser också att det är snabbaste vägen att testa ut hur metoden fungerar i praktisk drift.

En annan väsentlig orsak till att vi valde svarvningsoperationen är att det geometriska programmeringsproblemet kan betraktas som tvådimensionellt, och därigenom kan det behandlas något enklare än fleraxlig bearbetning.

3 METODBESKRIVNING

Mot den inledningsvis redovisade bakgrunden igångsattes ett projekt med huvudsyftet att hitta en förenklad metod för framställning av tillverkningsunderlag för de numeriskt styrda verktygsmaskinerna genom analys av påverkande faktorer.

Projektet har bedrivits som ett antal delprojekt som fört fram till en kravspecifikation av hur och med vilka medel man skulle vilja utföra programmeringsarbetet för NC-maskiner. Från denna kravspecifikation har projektet vidareutvecklats genom utarbetandet av en systemmodell, för hur man kan utnyttja ett interaktivt grafiskt datorsystem för att uppnå dessa mål. Projektarbetets olika moment framgår av fig 2.

3.1 KRONOLOGISKT SCHEMA

Vi ser i det kronologiska schemat hur de inledande arbetena med att analysera produktsortimentet och produktkomplexiteten kompletterades med ekonomiska analyser av programmeringsfunktionen och en kartläggning av den befintliga NC-organisationens för- och nackdelar. Tillsammans med den hårdvaruutredning som samtidigt genomfördes bildade dessa rapporter underlag för den STU-rapport som beskriver hur vi vill angripa programmeringsproblemet (STU-rapport 74-4121).

Genom att den i kravspecifikationen föreslagna datorutrustningen installerades på institutionen för mekanisk teknologi i Lund möjliggjordes det vidare arbetet på att förverkliga systemideéerna. Detta arbete har genomförts i olika etapper med något olika programmeringstekniska förutsättningar. Den första versionen byggde på en förenklad lösning som endast kunde användas för att behandla geometri och som aldrig kunde utgöra grund för de vidareutvecklingar omfattande verktygs- och skärdataval som anges i STU-rapporten. I den version som nu har tagits fram finns däremot dessa möjligheter med, och kommer att föras in successivt.

Förutom ovan angivna undersökningar som utförts vid maskinverkstaden har vi gjort ett antal studiebesök på ledande forskningsinstitutioner och industrier med stor omfattning av NC-bearbetning, för att studera hur man där har löst programmeringsproblemet. Genom vad vi sett av det produktionsförberedande arbetet under våra studieresor har vi stärkts i vår uppfattning att en förändring av teknik och metoder måste komma till stånd för att NC-tekniken skall kunna användas i full utsträckning även för småserietillverkning. Eftersom systemutveckling är mycket kostsam har ett stort antal av de internationellt sett största NC-teknikanvändarna skapat ett gemensamt diskussionsforum kallat CAM-I, Computer Aided Manufacturing - International. Inom denna organisation pågår ett kontinuerligt erfarenhetsutbyte och en gemensam forskning och utveckling av aktiviteterna kring NC-bearbetning. Genom medlemskap som educational members i CAM-I har vi kunnat följa dessa utvecklingssträvanden. Emellertid kan vi konstatera att de projekt som drivs inom CAM-I ofta riktar sig till verkstäder som är betydligt större och har en etablerad NC-verksamhet sedan mycket längre tid tillbaka än huvuddelen av den svenska industrin.

Vidare har vi studerat intressanta delar av litteraturen inom området, och därifrån hämtat idéer och uppslag. De verk som studerats men inte har någon direkt referens i avhandlingen har ändå medtagits i litteraturförteckningen för att utgöra referens för den som vill fördjupa sig inom området.

För att belysa bakgrunden till den i kapitel 4 beskrivna systemmiljön redovisas nedan några av de slutsatser som kunde dras av det tidiga utredningsmaterialet.

3.2 ANALYS AV DETALJKOMPLEXITET OCH BEARBETNINGSTEKNIK

Den huvudsakliga erfarenheten av NC-bearbetning i maskinverkstaden när projektet startade kom från bearbetningen av gjuteriets formrörsproduktion. Som ett första led i projektarbetet engagerades två examensarbetare från LTH för att försöka finna en beskrivande kod för det begränsade antalet rörtyper som förekommer med hänsyn till den av VIS-normer definierade placeringen av standarddetaljer (se fig 3), (Göransson, Roth 1974). På detta sätt ville vi finna en metod att ha färdiga programdelar för NC-bearbetning lagrade och kunna skapa program för unika rör bara genom att ange en kodbeteckning som beskriver vilka programdelar som skall ingå. Utredningen utmynnade i en beskrivningskod med 11 positioner, som förmår beskriva det relativt ensartade detaljspektrat.

Mot bakgrund av bl a framkomna resultat, modifierades målsättningen från att ha varit en gruppteknologiskt orienterad detaljstudie till att mera betona bearbetningstekniska likheter hos delar av detaljer. Med denna målsättning startade ett nytt examensarbete, denna gång med en stor del av maskinverkstadens svarvade detaljer som studieobjekt (Lettström, Månsson, Ralvert, 1974).

Den nya målsättningen var influerad av de system som vi studerat under de nämnda studieresorna, där särskilt demonstrationerna av bildskärmsteknik vid tekniska högskolan i Aachen och informationen om MITURN och dialog programmering vid TNO i Apeldoorn, givit idémässiga bidrag.

Ett viktigt resultat av den generella undersökningen av svarvade detaljer visade att det är svårt att hitta tillräckligt stora likheter mellan detaljerna som skall tillverkas för att man skall kunna återanvända gjorda program. En närmare undersökning visade emellertid att även om detaljerna var högst olika och ofta produktspecifika byggs den huvudsakliga formen upp av ett begränsat antal gemensamma segment. Genom att stapla bl a cylindriska, koniska och plana segment efter varandra längs centrumlinjen genereras en grov bild av detaljens geometri. Denna form modifieras sedan med produktspecifika segment så att detaljen får sin unika funktion.

Vi konstaterade således att man kan åstadkomma en preliminär beskrivning av geometrin med hjälp av ett litet antal gemensamma segment. Dessa segment är tillräckligt väldefinierade för att det skall vara möjligt att koppla ihop geometrisk form och dimen-

sion med den bearbetningsteknik som erfordras för att åstadkomma det unika segmentet. De geometribeskrivande segmenten kommer att kunna användas i automationshöjande syfte genom rutiner som föreslår verktyg för de olika deloperationerna. Segmenten kan därför betraktas som bearbetningsorienterade programmakron.

De nämnda produktspecifika segmenten beskriver sådan form som är unik för en viss produkt, t ex kolvringsspår. Denna form är oftast så speciell att man använder särskilda verktyg och bearbetningsmetoder och därför kan man även i det här fallet koppla ihop geometrisk beskrivning med verktygs- och metodval. Eftersom segmenten är direkt produktbeskrivande kan de liknas vid produktorienterade programmakron.

För att kunna beskriva maskinverkstadens omväxlande produkt-sortiment behövs således dels en uppsättning för alla detaljer gemensamma segment och dels för varje produkttyp speciella segment.

I ett försök att skaffa erfarenhetsmässiga bearbetningsdata till segmenten, genomfördes en undersökning av ett stort antal använda NC-program. Undersökningen omfattade en jämförelse mellan rekommenderade skärdataval och i praktiken använda skärdata samt ett försök till kartläggning av verktygsval och snittuppdelningsmetoder. Det mest påtagliga resultatet av undersökningen var hur kraftigt individorienterad den använda bearbetningstekniken är samt att yttre faktorer ofta begränsar verktygsleverantörernas optimistiska skärdatarekommendationer.

Man kan av dessa skäl inte säga att det existerar en enda riktig bearbetningsteknik för en viss detalj utan det finns flera val som var för sig kan motiveras (Hermansson, 1976). Detta senare konstaterande anser vi vara av stor betydelse för utformningen av programmodulerna för verktygs- och skärdataval. Utredningen styrker således uppfattningen att en dialog beslutsprocess där datasystemet ger alternativa förslag och där programmeraren beslutar, är den lämpligaste metoden att uppnå riktiga verktygsval och användbara skärdata. I sammanhanget bör nämnas möjligheten för maskinoperatören att anpassa varvtal och matningshastighet direkt över manöverpanelen, vilket kan ha påverkat programmerarnas skärdataval, vilket i sin tur gör det vanskligt att bedömma vad som är erfarenhetsmässiga skärdata.

3.3 JÄMFÖRANDE ANALYSER AV TILLVERKNINGSKOSTNADEN VID KONVENTIONELL BEARBETNING OCH NUMERISKT STYRD BEARBETNING

I vår strävan att bringa större klarhet i problematiken kring vilka detaljer som är lönsamma att tillverka i NC-maskiner genomfördes ett antal utredningar om vilka kostnadsposter som ändrades vid övergång till NC-bearbetning och hur dessa påverkar totalkostnaden för tillverkning.

De första frågeställningarna kan illustreras av bild 4. Denna bild visar konsekvensen av att byta ut en konventionell maskin mot en numeriskt styrd med högre maskintimkostnad och med bibehållen

operatörsinsats. Om man i första skedet bortser från att NC-maskinen kräver ett mera omfattande förberedelsearbete kan man konstatera att ett arbete som tager en viss tid att utföra i konventionella maskiner (A) kostar ett antal kronor (B) att tillverka. För att hålla kostnaden konstant får man vid NC-bearbetning (C) belägga produktionsenheten under väsentligt kortare tid (D). Eftersom NC-maskinen många gånger inte är särskilt mycket kraftigare än motsvarande konventionella maskin kan det vara svårt att klara denna produktivitetshöjning om man inte kan tjäna in tiden i form av minskade väntetider, ställtider eller verktygsväxlingstider.

När vi dessutom kunde konstatera att NC-tekniken för med sig omfattande produktionsförberedande arbete i form av programmering och verktygsförinställning beslöt vi genomföra en jämförande analys av ett antal produkter som tillverkats samt att taga fram en modell som kan användas för jämförelser av hur tillverkningskostnaden förändras vid övergång till NC-bearbetning (Bergqvist, Eriksson, 1975).

Genom omarbetning av en DIN-normerad kalkylmetod kom vi fram till en användbar modell för beräkning av skillnaden i tillverkningskostnad för NC-bearbetning och konventionell bearbetning. Den enskilda detaljens styckkostnad består av tre delar där en del utgör direkta särkostnader för detaljen, en del utgör gemensamma kostnader för tillverkningen av ett visst parti och slutligen en del består av initialkostnader för att man över huvud taget skall kunna tillverka detaljen. De senare kostnadsposterna skall sålunda divideras med partistorleken resp totala antalet tillverkade detaljer. Modellen får följande utseende:

$$\text{Styckkostnad} = \text{särkostnad} + \frac{\text{partikostnad}}{\text{partistorlek}} + \frac{\text{initialkostnad}}{\text{totalt antal detaljer}}$$

Till den enstaka detaljens särkostnader hör främst maskin- och operatörstimbekostnader som diskuterats ovan och som har svårt att ge ekonomisk fördel åt NC-bearbetningen. Av initialkostnaderna utgör programmeringen en väsentlig och oftast dominerande post. Eftersom den divideras med totala antalet detaljer kan man för långserieproduktion eller ständigt återkommande produktion acceptera relativt höga kostnader utan att de får en alltför stor negativ inverkan på tillverkningsstyckkostnaden.

Det är därför ytterst viktigt att man analyserar sitt detaljspektrum även med avseende på serielängd innan man beslutar om vilken programmeringsmetod som är mest lämplig för verkstadens produktion. Därför genomfördes och redovisades en undersökning där verktygsmaskinens totala planerade utnyttjandetid fördelats på de olika detaljtyperna som skulle bearbetas. På så vis erhöles ett antal huvudgrupper (horisontell indelning i fig 5 och fig 6). Inom var och en av dessa huvudgrupper uppskattades andelen detaljer av en viss seriestorlek i förhållande till andelen detaljer inom huvudgruppen och infördes som procentuella andelar. Av dessa båda figurer kan man sluta sig till att huvuddelen av maskinverkstadens produktion sker i serier under 30 detaljer,

vilket i sin tur visar vikten av att programmeringskostnaderna hålls låga.

Med vår önskan om att taga NC-tekniken i anspråk för tillverkning av relativt enkla detaljer i korta serier och partier har vi skapat en situation som kräver största uppmärksamhet för att inte resultera i ett oekonomiskt utnyttjande av NC-tekniken. Samtidigt kan vi konstatera att den beskrivna situationen är den vanligast förekommande inom allmän verkstadsindustri och därför måste metoder och lösningar tagas fram för en vidare expansion av NC-tekniken inom den mekaniska verkstadsindustrin.

3.4 KARTLÄGGNING AV BEFINTLIG NC-ORGANISATION

För att studera och följa upp vilken påverkan införandet av NC-teknik haft på produktionen i maskinverkstaden genomfördes en undersökning av vilka arbetsformer som utbildats kring NC-maskinerna (Holm, Svanberg, 1975). Målsättningen med undersökningen var att konstatera på vilket sätt uppdelningen av det traditionella maskinoperatörsarbetet i programmering, verktygsförinställning och maskinpassning har påverkat arbetsformerna och styrningsfunktionen i verkstaden.

Utredningen visade ganska snart att maskinoperatörens arbete i de flesta fall övergått i att passivt övervaka maskinerna. Besluten om bearbetningsmetod och verktygsval har flyttats bakåt i tillverkningskedjan och upplevs ibland som felaktiga men man saknar möjlighet att påverka programmeringen. Vidare kunde man konstatera att ledningsfunktionen för NC-verksamheten som borde ligga på arbetsledningen delvis överflyttats dit där programmeringskompetensen finns. En intressant iakttagelse i sammanhanget är vilka möjligheter som finns till en utvidgning av den påbörjade programmeringsverksamheten med vidareutbildade maskinoperatörer. En utvidgning av denna verksamhet dithän att alla NC-maskinoperatörer blir kapabla att programmera sina maskiner skulle ge den tillbaka den aktiva rollen i tillverkningen samtidigt med en naturlig vidareutveckling av arbetsuppgifterna.

Slutsatserna om hur tillverkningsekonomin skall kunna förbättras sammanfattades i en rapport "Metoder att uppnå bättre ekonomisk balans mellan NS-bearbetning och konventionell bearbetning" (Eriksson, 1975). Denna rapport kan ses som en katalog över de åtgärder som kan vidtagas och vilka krav som måste uppfyllas för att man skall uppnå förbättrad ekonomi vid NC-bearbetning. I första hand angriper vi problematiken med maskinoperatörskostnaden och programmeringskostnaden eftersom de är så dominerande.

Med en samtidig ökning av automationsgraden hos verktygsmaskinerna och ett förenklat programmeringsförfarande som även maskinoperatörerna behärskar, skulle man kunna åstadkomma nya intressanta arbetsformer för NC-bearbetning. Genom minskade krav på operatörsinsatser i verktygsmaskinerna kan en operatör betjäna mer än en maskin. Om den därvid frikopplade operatören kan sysselsättas med programmeringsarbete m h a ett metodorienterat programmeringssystem skulle lönsamheten med NC-bearbetning radikalt förändras. Som ett första steg i denna riktning skisserades ett

programmeringssystem för svarvning i rapporten IPS-system (Eriksson IPS, 1975). Tillsammans med den vid institutionen för mekanisk teknologi parallellt gjorda kartläggningen av vilka system och datorutrustningar som kunde tänkas användas, sammanfattades slutligen kravspecifikationsfasen i en rapport till STU. (STU - 74-4121.)

Det ovan summariskt sammanfattade materialet utgör grunden för den systemmiljöbeskrivning och systemmodell som diskuteras i de följande kapitlen.

4 SYSTEMMILJÖ

En av de viktigaste punkterna i beslutsgången vid investeringar i system och utrustning är att man noga kartlägger och dokumenterar den avsedda tillämpningen med avseende på alla påverkande faktorer. Till denna s k systemmiljö räknas inte bara den omedelbara omgivningens nuläge utan framför allt bör man försöka skapa förutsättningar för att systemet skall passa in i den typ av arbetsorganisation och de arbetsförhållanden som man tror kommer att gälla under dess livslängd.

De allmänna trender som idag är helt klara är exempelvis inriktning mot produktion med begränsad bemanning under delar av dygnet, en utökad tillverkning i produktionsgrupper med tillhörande för- enklade styrningsfunktioner samt allmänna krav på medinflytande och insyn i beredningsfrågor.

Nuläget och våra förutsägelser om systemmiljön i maskinverkstaden, som legat till grund för IPS-utvecklingen, redovisas i nedanstående avsnitt om produktsortiment, tillgänglig verktygsmaskinutrustning och NC-verksamhetens organisation.

4.1 PRODUKTSORTIMENT

Den produktmix som systemet skall kunna behandla framgår delvis av kap 3.2. Nedan redovisas den uppfattning om seriestorlekar, detaljkomplexitet och konstruktionsursprung som antages gälla för maskinverkstaden i framtiden.

4.1.1 Seriestorlek

Den befintliga produktionens sammansättning för support och karusellsvarvning framgår av bild 5 och 6. Längs horisontal-linjen anges de olika produkttyperna i avtagande proportion till totalvolymen. Staplarna är därefter indelade efter olika produkters serielängd där fältens yta speglar hur stor del av produkttypen som har den angivna seriestorleken.

Genomgående kan man säga att produkterna tillverkas i korta serier med huvuddelen i ett intervall mellan 5 och 30 detaljer. Den förväntade ökningen av legotillverkning och s k alternativproduktion kommer att medföra en ytterligare kortare serielängd. Denna utveckling förstärker vikten av att det framtida programmerings-systemet på ett ekonomiskt sätt kan behandla enstyckstillverkning (jfr kap 3.3).

4.1.2 Detaljkomplexitet

Utredningarna som nämndes i kap 3.2 visade att större delen av detaljsortimentet kunde beskrivas med ett begränsat antal grund-element. Det finns heller inga tecken som tyder på, att de nya produkter som det ändrade tillverkningsprogrammet vid varvet medfört, kommer att kräva mer komplicerade geometriska beskrivningsmodeller.

4.1.3 Konstruktionsursprung

Vid val av programmeringshjälpmedel är det mycket betydelsefullt att ritningsunderlagens ursprung utreds, d v s om det är frågan om egna eller externa konstruktioner. Betydelsen ligger främst inom följande tre områden.

- För egenproducerade ritningsunderlag kan måttsättning, toleransangivning, materialkoder etc anpassas till använt programmeringssystem. Att ställa särskilda krav på externt framställda ritningsunderlag är oftast omöjligt och därför måste man i sådana fall istället använda programmeringssystem med flexibelt indataformat.
- Om allt ritningsunderlag kommer från egna ritkontor kan man diskutera vilken geometrirepresentation som är lämpligast. Man finner kanske att bästa lösningen är att ritkontoren levererar detaljinformationen direkt i numerisk form. För att detta skall vara möjligt krävs det att man har en gemensam produktdefinierande geometri-beskrivning som gör den konventionella ritningen överflödig. Detta beskrivningssätt tacklas f n i ett stort antal internationella forskningsprojekt under beteckningar som "geometriska produktmodeller" och "geometric modelling". Förutom att finna ett gemensamt beskrivningssätt måste man hitta rationella metoder att taga hand om informationen och vidarebearbeta den till styrinformation för verktygsmaskinerna.
- I takt med att allt fler företag skaffar databaserade konstruktions- och produktionshjälpmedel finns det en möjlighet att utnyttja varandras underlag direkt för styrremsframställning eller bearbetning under förutsättning att en gemensam standard finns.

Om nu inte några förenklade förutsättningar föreligger utan den framtida produktionen och dess ursprung är mer eller mindre okänd måste man välja ett mångsidigt programmeringssystem med möjligheter att behandla ritningsunderlag med olika måttsättningstekniker på ett enkelt och snabbt sätt. Konverteringsarbetet av ritning till numerisk representation får inte vara den styrande faktorn vid val mellan konventionell och numeriskt styrd bearbetning.

För maskinverkstadens del kan man konstatera att en mycket stor del av ritningsunderlagen är externa konstruktioner (se fig 5 och 6. Från fartygsproduktionen kommer ett stort antal detaljer som konstruerats av turbin- och dieselmotorleverantörer. Från den alternativa produktionen kommer både egna konstruktioner och externa för de fall då varvet arbetar som "contractor" i ett projekt. Även de produkter som kommer från det egna gjuteriet eller smedjan för en efterföljande maskinbearbetning har skiftande ursprung.

En grov indelning av på marknaden förekommande system kan göras i följande kategorier.

- Generativa system där varje program byggs upp och betraktas som helt unikt. Denna typ används främst där man inte anser att det förekommer någon som helst anknytning mellan tillverkningspektrats detaljer.
- Generativa system med produktmakron. Möjligheten att skapa produktmakron kan utnyttjas för att framställa en företagsintern standard på vissa återkommande konstruktionsdetaljer. Metoden används inom företag där man kan urskilja vissa detaljfamiljer.
- Återvinningssystem. Som namnet antyder innebär det en ensartad produktion med mycket snarlika detaljer i ett fåtal varianter.

Med tanke på den stora mängden externa ritningsunderlag har vi bedömt det som osannolikt att maskinverkstaden skall kunna påverka ritningsutförandet i någon nämnvärd grad. Eftersom vi dessutom inte tycks kunna hitta tillräckliga volymer enhetlig produktion, som kan tillverkas enligt variantprincipen, är programmeringstekniken klar. Den enda möjliga tekniken är en generativ programmeringsmetod med generella geometrielement kombinerat med speciella produktmakron som beskriver direkt produktspecifika detaljer.

4.2 TILLGÄNGLIG VERKTYGSMASKINUTRUSTNING

IPS-metoden siktar på att tillvarata NC-teknikens olika möjligheter till rationell tillverkning. Vi vill därför börja med att införa rutinerna på en maskintyp där de manuella ingreppen från maskinoperatören, under bearbetningen av en detalj, är så få som möjligt.

Vi avser att starta med supportsvarven, eftersom den bl a har fördelarna av helautomatisk verktygsväxling med stort verktygsmagasin, god spånavgång genom snedbäddsutförande och automatisk borttransport av spånor. Bortsett från avsaknaden av automatisk växling av arbetsstycken och adaptiv styrning har maskinen goda förutsättningar att kunna utgöra en länk i ett framtida produktionssystem uppbyggt för att arbeta med begränsad bemanning.

En testverksamhet i supportsvarven är vidare förberedd genom att en examensarbetare engagerades under 1978 för att taga fram en postprocessor för konvertering av CL-information till håltremskod för Max-Müller supportsvarvar med Kongsberg styrsystem. Arbetet är slutfört och postprocessorn är testad på arbetsobjekt från maskinverkstaden.

Vi har även förberett en direktöverföring av håltremsinformationen från en minidatorbaserad programmeringscentral till supportsvarvens programminne. Arbetet, som utförts av Kockums Computer Systems AB under våren 1979, har haft som huvudsyfte att prova ut en lämplig transmissionsteknik för håltremsöverföring och BTR ingång till styrutrustningarnas programminne. De problem man därvid stöter på är framför allt beroende av överföringsavstånd, nöd-

vändig transmissionshastighet och störningskänslighet (Spur, Stule, Weck 1977). Den valda transmissionstekniken (RS 422) kommer att tas i provdrift med utnyttjande av en Kongsberg KS500 minidator för programadministration och en enkel programanrops-terminal vid maskinen. Denna lösning är emellertid endast tillfällig och överföringstekniken kommer därefter att utnyttjas mellan den programmeringsutrustning som placeras i maskinverkstaden och samtliga verktygsmaskiner.

Direktöverföring av hållemsinformation mellan programmeringscentral och verktygsmaskin kräver tillgång till stor lagringskapacitet på direktminne. Maskinverkstadens ca 1000 program kommer att kräva ett skivstationsutrymme i storleksordningen 40 M byte. Detta utrymmesbehov kan erhållas antingen genom att programmen lagras på någon av datacentralens skivstationer och utnyttjas över en terminal eller genom lagring av programmen på den till Computervisionsystemet hörande skivstationen.

Det är således väl sörjt för att introducera den nya programmeringstekniken och snabbt kunna dra nytta av decentraliserad eller distribuerad datakraft i maskinverkstaden. En andra etapp skulle innebära införande av IPS-rutinerna för programframställning till de båda karusellsvärvarna. Även dessa maskiner har automatiskt verktygsväxling med stora verktygsmagasin och automatisk spåntransport.

Att vi väljer dessa maskiner som etapp 2 beror på att det är frågan om tyngre maskiner där arbetsobjekten inte växlar i samma höga takt som i supportsvärven. Med supportsvärvens korta serietillverkningsstider och korta programlängd ger den oss bästa möjligheter att snabbt testa igenom systemet. Karusellsvärvarna har inte heller samma krav på snabbhet i rutinerna för omriggning, programinläsning och programuttestning som supportsvärven.

Detta gäller i än högre grad för axelsvärven, som med sitt extrema arbetsområde har ett såpass speciellt detaljspektrum, att flertalet tidigare nämnda faktorer har relativt liten betydelse.

De programkorrigeringsrutiner som kan användas i maskinverkstaden för manuellprogrammerade program är dels direkta ändringar medan programmet befinner sig i maskinstyrningen och dels remsrättningar över Flexowriter. Eftersom de korrigeringar som utföres direkt i verktygsmaskinen inte alltid leder till en omstansning av hållemsan förlorar man många av de förbättringar som maskinoperatörerna infört. Detta upplevs som otillfredsställande eftersom hållemsorna då alltid måste läsas in i verktygsmaskinerna och först korrigeras manuellt innan bearbetningen kan starta. Ett väsentligt krav på det nya systemet är därför att ISO- eller EIA-kodad hållemsinformation skall kunna lagras och editeras utanför maskinstyrningarna i samma utrustning som används för programmering och dataöverföring mellan programbibliotek och verktygsmaskiner.

4.3 NC-VERKSAMHETENS ORGANISATION

Den vid maskinverkstaden befintliga organisationen för NC-programmering kan betraktas som embryot till den arbetsmetodik som vi vill se tillämpad vid en utvidgad NC-verksamhet.

Genom att NC-organisationen är helt fristående från den konventionella beredningsavdelningen har man lyckats undvika att programmeringsarbetet flyttas längre och längre bakåt i organisationen. Denna trend är annars relativt vanlig i större företag där antalet NC-maskiner växer och där man allt eftersom programmeringsbehovet ökar satsar på mer och mer sofistikerade programmeringssystem. Dessa system kräver ofta större teoretiska kunskaper av användarna och programmeringsfunktionen vandrar bort från verkstaden.

När programmeringsfunktionen flyttas bort från verkstadsmiljön minskar möjligheterna till en successiv anpassning och på bearbetningserfarenheter byggd vidareutveckling av programmeringsrutinerna. Systemet permanentar således den teknik som gäller under utvecklingsskedet. Med den snabba utvecklingen, både på methodsidan och materialsidan är det ytterst viktigt att framtida system medger en direkt verkstadsanknuten systemuppdatering och detaljprogrammering.

För att uppnå denna informationsåterföring måste programmeringen integreras i maskinoperatörens arbetsuppgifter och ligga inom hans ansvarsområde.

Yrkesskicklighetens betydelse för ett gott programmeringsresultat understryks av att hela maskinverkstadens nytillskott av programmerare (4 man) togs ut bland intresserade maskinoperatörer. Dessa har som sin huvudsysselsättning att programmera men får då och då återgå till att vara maskinoperatörer. Denna växelvisa programmerings- och operatörstjänst borgar för att förbättringar på maskin- eller methodsidan får omedelbara effekter på nyprogrammeringen.

I brist på ett lättarbetat programmeringssystem används idag nästan uteslutande manuell programmering för svarvprogrammeringen. Detta har som följd att det tager lång tid att framställa program och tillverkningsunderlag och innebär dessutom att man måste ha nästan en programmerare för varje verktygsmaskin.

Det tidigare visade önskemålet om en verkstadsnära och enkel programmeringsmetod kommer vi att kunna införa genom IPS. När systemet fungerar i drift skall man betrakta programmeringsfunktionen som en naturlig del av maskinoperatörens arbete. Ett mål som vi vill uppnå med hjälp av systemet är således att återföra ansvaret för tillverkningen till maskinoperatören och genom att taga datatekniken till hjälp skapa ett produktionshjälpmedel som engagerar verkstadspersonalen att taga aktiv del i framställningen av tillverkningsunderlag. Därför kommer maskinoperatörerna successivt att få utnyttja programmeringssystemet för att själva programmera sina arbeten. Genom denna ökade självförsörjning av

programmeringstjänster lättar arbetsbelastningen på programmeringsfunktionen och vi kan utan att öka programmerarnas antal framställa underlag för fler detaljer. Detta skapar i sin tur förutsättningar för utnyttjande av NC-teknik för allt kortare tillverkningsserier.

En ökad förståelse av NC-tekniken i verkstaden och spridning av programmeringskunnandet kommer att minska behovet av programmeringsfunktions programmeringskapacitet. Kraven på den tekniska ledningen och programmeringsfunktionen kommer istället att öka när det gäller att tillhandahålla ständigt förnyad och aktuell databasinformation om verktygsmaskinkarakteristika, materialegenskaper, uppspänningshjälpmedel, verktygskartotek och skärdata. Tillsammans med ett metodorienterat programmeringssystem kommer detta att ge operatörerna den arbetsutvidgning och det inflytande över tillverkningsprocessen som vi åsyftar.

Avslutningsvis kan man konstatera att förutsättningarna och kraven som redovisats i ovanstående kapital inte leder fram till en entydig systemmodell. Det är bl a denna flertydighet som medför att företag med i grunden snarlika produktionsförutsättningar, har valt så olika programmeringssystem, och med fog kan hävda att det valda systemet är överlägset alla andra för den aktuella tillämpningen. I följande kapitel kommer först systemmodellen för IPS att beskrivas och därefter den tekniska systemutformningen.

5 SYSTEMMODELL

Avsikten med detta kapitel är att belysa hur IPS-systemet kommer att användas under produktionsmässiga förhållanden. Först beskrivs den tekniska uppbyggnaden av systemet, d v s vilken dator-teknisk utrustning som behövs för att åstadkomma en programmeringsplats enligt IPS idéerna. Den andra komponenten i systemmodellen är en generell diskussion om hur vi anser att programmeraren bäst skall kunna utnyttja den interaktiva grafiska tekniken. Slutligen kommer IPS-systemets olika funktioner att gås igenom mera i detalj. Tillsammans utgör de tre avsnitten en teknisk och programmässig modell, som uppfyller kraven på en programmeringsarbetsplats som framkom i analysen av systemmiljö.

5.1 ARBETSPLATSUTFORMNINGEN

Valet av datorteknisk utrustning beror till mycket stor del av i vilken utsträckning man vill att programmerarna skall kunna utnyttja datorstöd i arbetet.

En av målsättningarna med IPS är att datorstödet skall omfatta så många som möjligt av alla de uppgifter som utföres utan datorstöd i konventionella programmeringssystem, t ex katalogsökningar efter lämpliga verktyg. Vid detaljprogrammering av verktygsmaskiner har man konstaterat att dessa arbetsmoment ofta utgör 50 - 75 % av den totala programmeringstiden. Förutom att tiden för detaljprogrammering innehåller upp emot 75 % icke datorberoende programmeringsarbete går det åt mycket tid för administration av kataloginformation och programbibliotek. Så mycket som möjligt av detta arbete vill vi utföra i en programmeringsstation eller programmeringsarbetsplats.

För att underlätta förståelsen redogöres först i grova drag för datorsystemet som använts under utvecklingsarbetet. Därefter avhandlas hur man skall utnyttja det för programmering och programadministration samt vilka standardiseringseffekter man bör kunna uppnå. Detta ger i sin tur möjlighet att ställa upp ett antal krav på datorutrustning som måste ingå i ett produktionssystem.

5.1.1 Datorutrustning

För att genomföra IPS-idéerna installerades en interaktiv grafisk minidatorutrustning, så kallad CAD-utrustning, vid institutionen för mekanisk teknologi och verktygsmaskiner på tekniska högskolan i Lund. Eftersom CAD-system redan i standardutförande innehåller uttestade och samkörda rutiner för behandling av geometriska element ansåg vi en vidareutveckling av ett CAD-system vara den snabbaste vägen till vårt tillämpningssystem för svarvprogrammering.

Grundsystemet har levererats av Computervision Corp, USA och består av följande komponenter (se fig 7).

Centrala enheter

- minidator med primärminne 32k 16 bit
- skivminne 14 M ord
- magnetbandstation
- hålremsläsare, hålremsstans

Arbetsplatsutrustning

- grafisk bildskärm
- digitaliseringsbord med funktionstangenter
- termoskrivare
- alfanumeriskt tangentbord

Programvara

- CADDs, geometrisk grundprogramvara
- PEP, parameterdefinierade geometriska makron
- fortränkompilator
- tillämpningsprogrampaket
- källkod till hela systemet

Med ovan angivna utrustning har vi byggt upp ett tillämpningsprogram som motsvarar geometridelen som beskrivs i STU-rapporten 74-4121. IPS-idéerna kan givetvis även genomföras på andra interaktiva grafiska utrustningar. Beroende av hur det valda systemet är uppbyggt krävs emellertid större eller mindre ändringar i dialogens utförande.

5.1.2 Programmering

Under denna rubrik kommer endast arbetsplatsmässiga aspekter på programmering med IPS att behandlas. Dialogmässigt respektive funktionsmässigt behandlas programmeringen i kap 5.2 och 5.3 medan de olika ingående IPS-programmen redovisas i kap 6, tekniskt systemutförande.

Detaljprogrammering

På ett tidigt stadium i kravspecificeringsfasen av IPS-systemet konstaterades följande önskemål.

- Så stor del som möjligt av programmeringsarbetet bör utföras i direkt dialog med systemet.

En mycket stor del av programmeringstiden går idag åt till att slå i tabeller och pärmar. Det är uppgifter om ex vis tillgängliga uppspänningshjälpmedel, tillgängliga verktygsfästen, standardiserade verktygshållare, förrådsförda skärplattor, olika kvaliteter, rekommenderade nosradier vid olika ytjämnhetskrav, rekommenderade skärdata, matningar, varvtal etc. För att uttömmande behandla alla bearbetningsfall skulle listan över information bli lång. All denna information tages inte fram på en gång utan allt eftersom programmeraren behöver de olika bitarna i sitt program eller programmanus (vid utnyttjande av automatisk

programmering) sker en tabellslagning. Dessa sökningar återkommer således kontinuerligt under arbetets gång och innebär varje gång en störning i programmeringsprocessen.

De uppgifter som behövs är rena standarduppgifter och tillfällena då en viss informationsmängd behöver göras tillgänglig för systemanvändaren kan klart definieras. För att minska ovan nämnda störningar av programmeringsprocessen är IPS utformat så att beslutsunderlaget för varje enskilt programmeringsmoment kan åskådliggöras antingen i grafisk form på den grafiska bildskärmen eller som en textinformation på en alfanumerisk bildskärm. Programmering med hjälp av IPS innebär därför, att större delen av programmeringsarbetet utföres i en direkt dialog med datorsystemet.

För att förenkla programmeringsarbetet och kunna minska utbildningstiden för nya programmerare kunde vidare konstateras.

- Programmeringsdialogen måste vara så utformad att programmerarens successiva handlande hela tiden initieras av systemet.

Detaljprogrammering med hjälp av IPS kommer därför att kännetecknas av en systeminitierad dialog där stödinformation kan beordras fram på bildskärmen eller skrivaren allt eftersom programmeraren anser det nödvändigt. Vidare får inte avancerade programmanus vara nödvändiga före programmeringen kan påbörjas. Kraven på programmeraren inskränker sig till att han skall ha en idé om hur han vill bearbeta den aktuella detaljen.

Vid programmeringsarbetet skall han utnyttja den grafiska skärmen för att få stödinformation, för att kunna ange positioner och parametrar i förhållande till redan inmatad geometri samt för verifiering av erhållet grafiskt resultat.

Kontroll av program.

Uttestning av program kommer även i fortsättningen delvis att utföras i verktygsmaskinerna, men omfattningen av de test som måste utföras direkt i maskinen kommer att kunna minskas. Alla geometriska fel bör på sikt kunna sorteras bort redan i programmeringsskedet genom att man har tillgång till grafisk representation av det numeriska underlaget. Det är i princip endast de fel som genereras av post-processorerna som inte kan kontrolleras i programmeringsutrustningen.

Editering av program.

Under rubriken editering av program avses direkt eller indirekt rättning av felaktig "hålremsinformation". När detaljprogrammeringen anses klar, dvs geometrier och bearbetningsdata teoretiskt verkar riktiga, behandlas denna information i en postprocessor för generering av styrinterinformation för en viss bestämd verktygsmaskin. Denna styrinterinformation stansas vanligtvis på hålremsa som sedan sättes i remsläsaren på verktygsmaskinen som därmed är klar för programtestning.

De fel som dyker upp vid denna testprocedur kan rättas på två olika sätt, nämligen

- 1) genom att informationen före postprocessorkörningen, d v s CL-filen rättas och att därefter den nya CL-filen behandlas i postprocessorn
- 2) genom att man editerar, d v s ändrar, tillfogar eller raderar programblock i "hålremsinformationen".

Vilken metod man bör välja kan vara svårt att bestämma eftersom man i första fallet gör en indirekt rättning, vilket kan generera nya fel vid den efterföljande postprocessorbehandlingen. Denna metod ger möjlighet till verifiering på bildskärm genom att ändringen göres i den databas som kan representeras på Computer-vision systemets grafiska bildskärm. En direkt rättning däremot eller editering av "hålremsinformationen" har sin fördel i att man slipper utnyttja IPS-programmen, CADDS-programmen eller postprocessorer. Här går man å andra sidan miste om möjligheten att verifiera den gjorda rättningen på den grafiska bildskärmen.

Genom att IPS-systemet med sina grafiska verifieringsmöjligheter förväntas ge relativt få fel och endast smärre justeringar av bearbetningsdata kommer att vara nödvändiga, bör man satsa på direkträttning av "hålremsinformation". Denna rutin kan då dessutom ge kortare test- och inkörningstider av hålremsor i verktygsmaskinerna. Tid som kan användas till produktionskörning i verktygsmaskinerna i stället för till programkorrektur.

Computervisionssystemet innehåller i sitt grundutförande rutiner som är mycket användbara för editering av direkt "hålremsinformation". Det finns således ett kommando EIAIN som innebär inläsning av EIA-kodad hålremsa till en datafil som lämpligtvis betecknas med hålremsnumret. Denna fil är därefter editierbar med systemets alla editeringskommandon.

När det nya programmet är klart kan det stansas ut omedelbart över hålremsstans med hjälp av kommandot EIAOUT.

Tillgången till den fulla editeringsmöjligheten har så stor praktisk betydelse att den inte får förbigås. Ett av editeringskommandona heter X-tract och innebär att man på ett mycket enkelt sätt kan sätta ihop ett nytt program av olika delar av redan gjorda program. Man anger helt enkelt vilka rader ur ett godtyckligt program i databasen som man vill ha infört i det nya programmet. Dels kan man då plocka avsnitt ur program som man använt i produktion och dels kan man skapa generella programdelar för återkommande problem och sedan utnyttja dem på samma sätt som s k makrobibliotek.

Det är således tre huvudtyper av programmeringsarbete som skall utföras i programmeringsstationen.

- detaljprogrammering

- kontroll och verifiering
- editering av hållemsor

Utöver dessa rent programproducerande aktiviteter skall systemet utnyttjas för programadministration som beskrivs i nästa kapitel. Dessutom räknar vi med väsentliga produktionskostnadsinbesparingar genom de standardiseringseffekter som behandlas i kap 5.1.4.

5.1.3 Programadministration

Med programadministration avses vanligtvis bara själva hållems-hanteringen. Eftersom IPS-systemet förutsätter tillgång till egen datorkapacitet och systemet är uppbyggt för att hantera bildinformation såväl som textfiler bör en vidareutvecklad programadministration innehålla:

- Programbibliotek inkl erforderlig programdokumentation som grafisk representation, uppspänningsskisser, verktygsdata med ställängdsinformation, hållemsinformation etc.
- Möjlighet att direkt överföra hållemsinformationen från programmeringsarbetsplatsen till verktygsmaskinen över fasta ledningar.
- Möjlighet att återvinna gjorda program med gruppteknologiska metoder.

Programbibliotek.

I jämförelse med de flesta förekommande system innebär ett IPS-orienterat programsystem en del nya möjligheter att utnyttja sparad information.

Eftersom den enda information som är intressant för maskinoperatören är hållemsinformationen och en skiss över vad som ingår i operationen är det också allt som ingår i biblioteket. IPS-systemet lagrar således inte undan några programlistor från genereringsfasen förutom den sk CL-filen (CL = Cutter Location), och därför består programbiblioteket av en geometrisk representation av rå och finkontur, verktygsvägar i form av CL-filen och slutligen en direkt kopia av hållemsinformationen. I den geometriska representationen av konturer och verktygsvägar finns tilläggsinformation om uppspänningar, verktyg och skärdata. Sambandet mellan IPS-systemet, den geometriska representationen, CL-filen och hållemsinformationen framgår av fig 8.

I den geometriska representationen kommer i ett vidareutvecklat skede förutom konturer och verktygsvägar även uppspänningsdon som chuck, stöddocka och pinol att ingå. På detta vis får man en bättre uppfattning om hur detaljerna bearbetas.

En fördel med Computervisionsystemet är att redan befintliga hållemsor kan tillföras programbiblioteket relativt lätt. Genom de i kapitel 5.1.2 nämnda rutinerna EIAIN och EIAOUT kan maskinverk-

stadens befintliga remsarkiv läsas in och administreras på samma sätt som nyprogrammerade detaljers EIA-format.

Figur 9 visar informationsflödet vid programmeringsförloppet och anger i vilket format de olika skedena lagras och vilka sökbegrepp som används för att definiera ett visst program.

Direktöverföring av håltremsinformation

Eftersom håltremsläsarna på de numeriskt styrda verktygsmaskinerna är den vanligaste felkällan vid NC-bearbetning har man på många håll intresserat sig för att direktöverföra håltremsinformationen från en lokal programmeringsstation till verktygsmaskinstyrningarna (Dietz 1977). Denna teknik att koppla förbi remsläsaren kallas BTR (Behind the Tape Reader) och är den idag mest använda metoden vid s k DNC (Spur, Stute, Weck 1977). DNC = Direct Numerical Control.

På maskinverkstadens uppdrag bedrivs för närvarande en försöksverksamhet av KCS (Kockums Computer Systems AB) med syftet att hitta den lämpligaste överföringstekniken för håltremsinformation från en minidator till verktygsmaskinstyrningarna. Den i dessa försök funna tekniken kan sedan komma att utnyttjas för en koppling mellan en framtida minidatorbaserad programmeringsstation och verktygsmaskinerna.

Programåtervinning

Vi konstaterade redan i analyskedet av maskinverkstadens detaljsortiment att produktionen var alltför heterogen för att ett grupp-teknologiskt återvinningssystem skulle vara rätt metod att för- enkla programmeringsarbetet. Denna uppfattning är säkerligen riktig men det kan för den skull ändå vara av intresse att ha tillgång till någon form av grupp-teknologisk indelning av detaljerna.

I framtida verkstäder kommer grupp-tillverkning säkert att ha relativt stor omfattning (Nya Fabriker, 1978) och då behövs det ett grupp-teknologiskt system för indelning i tillverkningsgrupper. Detta system behöver inte alls ha samma precision som ett system för programåtervinning och därigenom bör antalet kodsiffror kunna reduceras jämfört med de system vi studerade.

Åter igen ser vi möjligheten att utnyttja befintliga systemrutiner i det för IPS använda systemet. Genom att tillfoga en grupp-teknologisk kod som s k partproperty till varje inlagd programgeometri har man skapat en klassningskod. Denna kod kan sedan utnyttjas av biblioteksrutinerna för sökningar och listningar av närbesläktade detaljer i hela databasen.

5.1.4 Metod och urvalsstandardisering

En väsentlig del av programmeringsarbetet består som tidigare nämnts av att utnyttja tabell och kataloguppgifter. Med den i tidigare kapitel angivna datorstödsmodellen kommer nödvändig och tillräcklig information alltid att kunna presenteras över bildskärmen eller som en textutskrift på termoskrivaren.

Genom att utnyttja datorsystemet för presentation av katalog-information får man en obruten programmeringsprocess där varje beslutssituation för programmeraren åtföljs av informationspresentation. Förutom att programmeringsarbetet då går snabbare vinner man en fördel genom att alla programmerare utnyttjar samma beslutsunderlag för skärdata och verktygsval, vilket ger jämnare programkvalitet.

De idag använda programmeringsmetoderna ger visserligen ett friare val av skärdata och verktyg, men denna frihet betalas dyrt genom att mer tid förbrukas för t ex val mellan likvärdiga verktygsalternativ. Samma sak gäller för val av skärdata. I det konventionella programmeringsarbetet utnyttjas rekommendationer från tillverkare av skärplattor eller forskningsinstitut, justerade mot erfarenhetsmässiga reduceringar beroende av ex vis en enskild maskins stabilitet. När programmeringen utföres manuellt dokumenteras inte alltid dessa erfarenheter, som bygger på personalens yrkesskicklighet utan var och en som börjar programmera måste lära sig sin maskin, sina verktyg, sitt detaljspektrum och därifrån skapa sina personliga erfarenhetsvärden. Detta leder till att det tar lång tid innan en programmerare blir duktig i att optimera bearbetningsgången, samt att programkvaliteten blir mycket varierande beroende av om det är en erfaren eller ny programmerare som gjort arbetet.

Så länge man arbetar med ett litet antal programmerare kanske olägenheterna inte är så stora men när antalet numeriskt styrda maskiner ökar kommer behovet av programmeringskapacitet att öka snabbare än tillgången på erfarna programmerare. Då är det mycket värdefullt om man kan ge de mindre erfarna programmerarna tillgång till god yrkesskicklighet genom att presentera verkstadsanpassat datorstöd.

Samtidigt bör man kunna uppnå stora standardiseringsvinster genom att man begränsar antalet tillåtna verktygstyper och skärplatteskvaliteter. Kostnaderna för förrådshållning av ett stort antal specialverktyg eller särskilda skärplatteskvaliteter överstiger oftast den inbesparing man kan göra om man lyckas optimera valet för varje enskild deloperation. Valet blir dessutom snabbare om antalet alternativ minskas.

För att uppnå de förespeglade standardiseringsvinsterna och den rationella programhanteringen krävs det att systemansvaret för programmeringsstationen är lagt på den tekniska verkstadsledningen. Genom kontinuerlig uppföljning av olika metoders resultat kan man anpassa systemförutsättningarna efter de förändringar som nya material och ny bearbetningsteknologi medför. En beslutad åtgärd kommer därefter att kunna få omedelbara effekter på all nyprogrammering. Dessutom kan all information som lagras som tilläggsinformation till element i databasen bytas ut över hela programbiblioteket med enkla kommandon. Detta möjliggör ex vis utbyte av skärplatteskvaliteter genomgående i alla program med computervisions kommandon DATA EXTRACT och DATA MERGE.

Med detaljprogrammeringen konsekvent genomförd över systemet får man betydligt bättre och snabbare hantering av det växande och allt mer svårhanterade programbiblioteket.

5.1.5 Programmeringsstationens utrustning

När nu omfattningen av arbetet som på sikt skall kunna utföras i programmeringsstationen redovisats kan kraven på datorutrustning för en programmeringsverksamhet med maskinverkstadens omfattning och struktur sammanfattas enligt figur 10.

Jämfört med systemet som använts för programutveckling (se fig 7 har följande ändrats.

Centrala enheter

- Minidatorns primärminne utökas med 2⁴ K ord per terminal. Totalt 80 K ord för tre terminaler.
- Skivminnena till systemet är idag 40 M ord stora. Detta medför att tillräckligt utrymme finns för lagring av maskinverkstadens programbibliotek.
- Radskrivare för utskrift av programlistor åt maskinoperatörerna bör ingå.
- Plotter för dokumentation av grafisk information bör ingå.

Arbetsplatsutrustning

Två typer av arbetsplatser bör finnas.

Terminaltyp 1:

För nyprogrammering av detaljer behövs två terminaler med både grafisk och alfanumerisk bildskärm. Således bör termoskrivaren i standardutrustningen bytas ut mot en alfanumerisk bildskärm, s k textskärm. Detta är helt nödvändigt i ett produktionssystem där utskrift över den "oändliga" termoskrivarremsan dels är långsam, dels ofta är svårläst, förutom att den information som tages ut inte skall arkiveras.

Terminaltyp 2:

Den andra terminaltypen består av enbart alfanumerisk bildskärm och används för programadministration och programredigering. För dessa arbetsuppgifter är det oftast helt ointressant att visa grafisk information och därför är en enklare terminal tillräcklig.

Eftersom terminaltyp 1 även kan användas för programadministration och programredigering men inte tvärt om kan man i ett inledningsskede slopa terminaltyp 2. När sedan omfattningen av systemadministration, editering och dataöverföring ökar kommer behovet av terminaltid för ren filhantering att bli så stort att en separat alfanumerisk bildskärm behövs.

Anläggningar av denna omfattning har anskaffats av mellan tio och tjugo svenska företag att användas huvudsakligen för datorstödd konstruktion.

5.2 DIALOGUTFORMNING

Mot bakgrund av systemmiljön som IPS skall användas i och kriterierna på god systemdialog (Martin , 1973) har en s k systeminitierande programmeringsdialog valts. Denna dialogtyp ger två stora fördelar när programmeraren är mer praktiskt än analytiskt lagd.

En systeminitierad dialog kräver inte samma förberedelsearbete i form av genomarbetade abstrakta manus, som en operatörsinitierad dialog. Det är således programmet som håller reda på den logiskt riktiga arbetsgången och därför kräver inte detta arbetssätt att programmeraren håller alla programmeringsdetaljer i minnet på samma vis som vid konventionell programmering. Programmeraren blir mindre känslig för störande moment, som t ex telefoner som ringer, maskinoperatörer som har bearbetningstekniska frågor, och kan bryta arbetet mitt i ett program utan att för den skull behöva flera timmars arbete för att rekapitulera gjort arbete och kunna slutföra programmeringsuppgiften.

Den andra fördelen med en systeminitierad dialog är att alla programmeringsuppgifter blir lösta efter samma principer oberoende av vem som är programmerare. Programkvaliteten blir därför jämnare och denna synpunkt kommer att bli allt viktigare ju fler icke specialister som skall programmera.

Eftersom denna typ av programmering inte utnyttjar datorn med kringutrustning lika effektivt som vid stora satsvisa datorkörningar måste man räkna med högre kostnader för datortid. De tidigare nämnda fördelarna motiverar emellertid trots det högre priset den systeminitierade dialogtypen. Erfarenheterna i maskinverkstaden av att använda programmerare med olika teoretisk bakgrund visar också att en systeminitierad dialog är att föredra, genom att den utjämnar programmerarnas olika programmeringskunnande.

De programmeringstekniska lösningarna av dialogen redovisas i kap 6 medan de bakomliggande tankarna om interaktivitet, informationspresentation och automatiseringsgrad behandlas nedan.

5.2.1 Interaktivitet

Som tidigare nämnts är dialogen uppbyggd av ett logiskt besluts-träd där systemet frågar programmeraren om uppgifter som behövs för att generera de olika bearbetningssegmenten. Dessa beskriver dels detaljen geometriskt och dels anger de villkor för bearbetningstekniken.

De svar systemet kräver är av följande tre slag.

- Val av alternativ ur en presenterad lista eller grafisk bild.
- Ja eller nej på en presenterad fråga eller förslag.
- Koordinatangivelse eller annat numeriskt värde på parametrar.

Systemet behandlar de avgivna svaren och går vidare i besluts-trädet genom nya frågor till programmeraren. Den eventuella stressituation som kan uppstå hos programmeraren genom att systemet i stort sett omedelbart efter ett svar från programmeraren ställer en ny fråga, anser vi är eliminerad genom att det alltid finns ett beslutsunderlag tillgängligt.

5.2.2 Informationspresentation

Detta beslutsunderlag kan visas på olika sätt beroende av dels programmerarens behov av datorstöd och dels vilken typ av underlag det gäller. Många av de dialoga system som fanns utvecklade i början av sjuttiotalet har en stel dialog som varken är avpassad till nybörjarens behov av stort datorstöd eller den mycket erfarne programmerarens önskan om en kompakt dialog.

Medan programmeraren ännu är oerfaren kan han med IPS-systemet begära grafiskt stöd över bildskärmen till varje beslutssituation för att därefter successivt övergå till minnesstöd i form av textinformation över skrivaren. När programmeraren slutligen känner systemet väl och kan följa dialogen helt utan stödinformation har vi uppnått det kompakta skrivsätt som ger lägst programmeringskostnader och som passar den rutinerade programmeraren bäst. Genom att man kan bestämma omfattningen av datorstödet separat i varje programmeringssteg är dialogen mycket flexibel och bör kunna tilltala alla användarkategorier. Detta är särskilt viktigt i en systemmiljö där programmerarnas kunnande och erfarenheter är så skiftande som beskrevs i kap 4.3.

Grafiskt bildstöd

Det grafiska bildstödet åstadkommes genom att stödbilder som finns inlagda i systemet är uppdelade på s k layer och kan visas oberoende av varandra. Vid de tillfällen då operatören kan tänkas behöva grafiskt bildstöd ges en möjlighet att göra de layer som innehåller lämplig stödinformation synliga. Stödinformationen kan bestå av segmentmenyer, beskrivningar av parameterfältet för enstaka segment, verktygsmenyer, beskrivningar av uppspänningsanordningar etc.

Förutom tidigare nämnda effekter på standardisering och uttestning av nya bearbetningsmetoder medför IPS-dialogens lättillgängliga datorstöd att programmeraren aldrig behöver avbryta en påbörjad programmeringsuppgift för att söka efter katalogdata eller liknande. Idag innebär programmering en sönderhackad följd av programmeringsarbete och informationssökning, vilket leder till längre total programmeringstid och sämre programmeringseffektivitet.

Textinformation

I de fall där grafisk information inte behövs kan programmeraren begära en textutskrift som minnesstöd som ex vis en verktygslista, maskinbegränsningar, skärdatarekommendationer etc. Att presentera stödinformation som textutskrift sker nära nog momentant och där-

för blir programmeringsarbetet något snabbare än med fullt grafiskt bildskärmsstöd.

Direkt dialog

Den snabbaste programmeringsmetoden är naturligtvis när programmeraren kan systemet tillräckligt väl för att kunna besvara dialogens frågor direkt utan datorstöd. Man kan förutsätta att systemet kommer att användas som en direkt dialog så snart produktionen är något sånär ensartad medan behovet av datorstöd ökas när nya produkter, ny maskinutrustning eller nya produktionsförutsättningar införes.

En fördel med IPS-dialogens flexibla datorstöd är att man underlättar systemförändringar och införandet av nya funktioner i systemet. Genom det omfattande datorstödet med bild- och textinformation kan man utöka det igång varande systemet utan stora satsningar på speciell utbildning och träning. Varje programmerare kommer i stället att kunna avpassa datorstödet efter sitt eget behov av hjälp. I en framtid när allt fler människor kommer att utnyttja datateknik i sitt arbete blir denna synpunkt allt viktigare, således inte bara för programmering av verktygsmaskiner.

5.2.3 Automatiseringsgrad

Nästan lika gammal som NC-tekniken är strävan att uppnå sk automatisk programmering. De banbrytande arbetena ledde så småningom fram till APT-språket (Automatically Programmed Tools) som idag kan sägas utgöra en inofficiell standard. Grovt karakteriserat kan utvecklingen av APT och dess utvidgningar och moderniseringar betecknas som en strävan efter att uppnå en mycket hög automatik genom stora och komplicerade rutiner. Denna utvecklingsväg är till mycket stor del framtvingad av de datorer och datatekniska metoder som fanns tillgängliga under systemens utvecklingsperiod.

Med ett systems automatiseringsnivå avser man ett mått på hur många av programmeringsarbetets aktiviteter som kan utföras med datorstöd (Holmer, IVF-resultat 78649). Som framgått av kapitel 5.1.2 strävar vi efter att uppnå en mycket hög automatiseringsnivå med IPS-systemet. För en bestämd aktivitet i systemet kan därefter automatiseringsgraden vara olika hög. Automatiseringsgraden beror på i vilken omfattning systemet har tagit över besluten från programmeraren.

Genom att IPS-systemet utnyttjar en interaktiv programmeringsteknik med en omfattande hjälpinformation behöver automatiseringsgraden i systemet inte vara lika hög som i system med satsvis bearbetning. Därigenom har programrutinerna kunnat göras väsentligt enklare samtidigt som körtiderna i datorn blivit kortare, vilket i sin tur möjliggjort användandet av en minidator i stället för stor dator. Målsättningen vid dialogutformningen har varit att få fram ett system som tar tillvara det bearbetningstekniska kunnandet hos verkstadspersonalen och möjliggör en enkel programmeringsteknik.

5.3 FUNKTIONSUTFORMNING

De IPS-rutiner som tagits fram består dels av rutiner som hjälper programmeraren till en logiskt riktig programmeringsföljd och dels av rutiner som behandlar de enskilda geometrisegmenten. Integrationsnivån eller automatiseringsnivån framgår av fig 11 som schemamässigt visar systemdialogen.

Nedanstående är en beskrivning över aktiviteterna som programmeraren skall utföra och vilka alternativ han kan välja mellan när han arbetar med IPS-rutinerna. För varje moment redovisas dels de olika valmöjligheterna programmerarna har till sitt förfogande och dels vilket datorstöd han har tillgång till för att fatta sina beslut.

Vi förutsätter att programmeringsstationen har hög utnyttjandegrad och att datorutrustningen har startats av den person som är systemansvarig i verkstaden. När detaljprogrammeringen skall börja kommer programmeraren till den grafiska terminalen med en ritning över arbetsstycket.

Systemstart

Första steget vid programmering är alltid att man identifierar arbetet som skall påbörjas. Detta identitetsbegrepp kommer sedan att användas för all information som genereras i samband med det aktuella arbetet, och utgör det huvudsakliga sökbegreppet i databasen tills informationen har passerat en postprocessor och övergått i hålremsinformation. Det är därför mest lämpligt att använda ritningsnumret som identitetsbegrepp fram till postprocessorbearbetningen och först därefter utnyttja programnummerbeteckningar.

Programmerarens första uppgift blir således att mata in ritningsnumret över tangentbordet.

För att starta IPS-dialogen och få tillgång till stödinformation indikeras funktionstangenten märkt START.

Dialogsteg 1

Genom indikeringen startar en dialog där systemet begär in uppgifter som behövs i de efterföljande programmodulerna för generering av produktmodellen eller för de programhanterande biblioteksrutinerna. På termoskrivaren skrivs följande begäran ut.

- 1) Ange material.
- 2) Ange ritningsbenämning eller kod.

Eftersom det är fullt möjligt att en detalj skall bearbetas i mer än en typ av NC-maskin bör man redan på detta stadium ge produktmodellen materialinformation. Denna information lagras som part-property och kan sedan utnyttjas som ingångsparameter för skärdatamodulerna.

Den idag ofta använda registreringen av ritningsbenämningen är inget tillförlitligt sökbegrepp om man vill leta fram likartade detaljer ur ett ritnings- eller programbibliotek. Orsaken är att ritningsbenämningen inte enbart beror på vad den enskilde konstruktören velat kalla detaljen utan dessutom på i vilken produkt den ingår. Det är inte ovanligt att i princip likartade detaljer kan ha olika benämning i olika produkter. På sikt vill vi därför byta ut ritningsbenämningen mot en detaljbeskrivande kod så att vi framtiden har en detaljanknuten gruppteknologisk klassificeringskod för informationsåtervinning.

Dialogsteg 2

Detta steg innebär val mellan att använda IPS-rutinerna för svarvning eller standard CADDS-rutiner för borrarning och fräsning. För den fortsatta beskrivningen antar vi att IPS-rutiner för svarvning väljs. Menyn över alternativen har följande utseende och de olika alternativen väljs genom inmatning av tillhörande kodsiffra.

- 1) Skriv ut lista över alternativ.
- 2) Svarvning.
- 3) Borrarning eller fräsning.

Efter att ha använt systemet en tid lär man sig kodbeteckningarna för de olika aktiviteterna utantill och behöver inte datorstöd för att ange rätt, men om programmeraren är ny ger han kodsiffran 1 över tangentbordet. Genomgående för hela IPS- och CADDS-systemet betyder kodsiffran 1 att man vill ha utskrivet en lista över de alternativ som finns att välja mellan.

Dialogsteg 3

När programmeraren matat in kodsiffran 2 är det klart att det är någon av aktiviteterna kring svarvningsoperationen som skall utföras. På samma sätt som för de olika bearbetningstyperna är de olika svarvningsaktiviteterna kodade och man kan välja mellan att ange koden direkt eller få listan med alternativ utskriven genom att mata in en etta. Denna lista innehåller följande information.

- 1 Skriv ut lista över alternativ.
- 2 Geometribeskrivning.
- 3 Verktygsdefinition.
- 4 Verktygsväggsgenerering.
- 5 Tilldelning av skärdata.
- 6 Postprocessorkörning av CL-data.

Dialogsteg 4

Genom att ange kodsiffran 2 har systemet fått tillräcklig information för att veta att en svarvad detalj i det angivna materialet skall geometribeskrivas. Dialogen har fört fram till den egentliga geometriska beskrivningen och systemet begär information om den aktuella detaljens totallängd och maximala diameter. Det är uppgifter som behövs för att bestämma en lämplig skalfaktor så att hela detaljen ryms på bildskärmen.

Eftersom vi har valt att utföra geometribeskrivningen i första kvadranten för alla typer av svarvbearbetning behöver programmeraren inte fundera över några teckenkonventioner utan kan använda koordinaterna direkt som de står på ritningen. För supportsvarven är det riktigt att arbeta i första kvadranten även bearbetningstekniskt eftersom den har en bakomliggande verktygshållare. För karusellsvarvarna och axelsvarven låter vi datorn beräkna spegelbilder i systemets Y-axel och X-axel innan verktygsväggsgenereringen startar.

Dialogsteg 5

Fram till detta stadium av dialogen anser vi att man klarar sig med enbart textinformation som minnesstöd, medan det är i den kommande geometridefinitionen som den grafiska hjälpinformationen behövs.

För att få tillgång till bildinformationen hämtar systemet en stödbild från skivstationen och överlagrar den på den ännu så länge tomma datarepresentationen av ritningsgeometrin. Genom att den överlagrade stödbilden är uppdelad på olika synlighetsnivåer (layer) och det totalt kan definieras 255 layer kan upp till 255 stödbilder hanteras av systemet. För den rena geometrihanteringen utnyttjar IPS-systemet ca 40 layer och för verktygskartoteket räknar vi med ca 20 layer. Av de återstående kommer vi på sikt att utnyttja ett 20-tal för att visa en grov kontur av svarvens, chuckens, stöddockans, verktygsmagasinet och dubbdockans geometri.

När stödbilden är inläst kan den numeriska beskrivningen av detaljen börja. För att underlätta verktygsväggsgenereringen i ett senare skede delas geometrin principiellt upp i yttre- och innerkontur för råämne och färdig detalj. Eftersom uppdelningen av vilken del av detaljens hela kontur som skall definieras som innerkontur och vilken del som skall kallas ytterkontur delvis beror på i vilken maskintyp man avser bearbeta detaljen är detta steg svårt att automatisera. Genom att låta programmeraren dela upp geometribeskrivningen får vi en tillverkningsriktig uppdelning redan vid genereringen.

Programmeraren får på samma sätt som i tidigare steg ett antal kodade alternativ att välja mellan. Menyn för steg 5 ser ut på följande vis.

- | | |
|---|---------------------------------|
| 1 | Skriv ut lista över alternativ. |
| 2 | Råämne ytterkontur. |
| 3 | Råämne innerkontur. |
| 4 | Färdig detalj ytterkontur. |
| 5 | Färdig detalj innerkontur. |

Den av programmeraren avgivna kodsiffran medför en aktivering av ett till konturtypen tilldelat arbetslayer. Detta innebär att alla geometrielement som genereras, som tillhörande konturtypen, lägges på tilldelat layer och därigenom blir lätt identifierbara.

Som framgår av fig 11 har vi byggt upp ett programhuvud som be-

står av följande partproperties.

- Ritningsnummer
- Material
- Grupptechn kod
- Svarvningsoperation
- Geometribeskrivning
- Skalfaktor/dimensioner
- Konturtyp

Dialogsteg 6

Dialogen har nått fram till den punkt där programmeraren börjar använda systemet för att avkoda ritningen. Genom nedbrytning av svarvningsoperationen för ett stort antal detaljer fann vi i kap 3 att man bearbetningstekniskt kan skilja ut ett begränsat antal segment som förmår beskriva flertalet detaljer. Segmenten kan indelas i huvudsegment som beskriver detaljen i grova drag, modifierande segment som finjusterar geometrin genom avrundningar, hålkälar etc, samt produktspecifika segment som tillfogar geometri som är direkt produktberoende och då ofta kräver specialverktyg.

Av bild 12 framgår vilka huvudsegment och modifierande segment som behövs för att beskriva en generell detalj.

Det sjätte steget i dialogen kan betraktas som en slinga där varje fullbordat varv medför att ett nytt segment bestämts, definierats och tillfogats databasen. Slingans omfattning framgår av fig 11.

Eftersom det ofta är den numeriska definitionsmodellen som är besvärlig i programmeringssystem har vi byggt in möjligheten att få grafiskt stöd på flera ställen.

Systemet börjar med att begära att programmeraren anger vilken av de tre segmenttyperna han vill använda. Nästa steg är att han skall ange segmentet han vill arbeta med. I båda valsituationerna ges han en möjlighet att begära bildstöd i form av menyer på bildskärmen.

De olika segmenten är kodade på motsvarande vis som beskrivits för den tidigare dialogen. Genom inmatning av en kodsiffra väljs segmentet och systemet svarar genom att presentera den aktuella parameteruppsättningen. Genom jämförelser mellan måttsättningen på ritningsunderlager och parameteruppsättningen för segmentet kan programmeraren mycket snabbt mata in parametervärdena till systemet. På samma vis som vid segmentvalet kan programmeraren begära bildhjälp och får då en detaljerad bild över segmentet och dess parametriska beskrivning. Vidare är programmen som definierar segmenten skrivna så att de för varje nyinmatat parametervärde testat om segmentet kan definieras. På detta sätt underlättas en snabb avkodning av ritningen. När segmentet är definierat visas det på bildskärmen och skrivs i databasen. I samband med att segmentet definieras ges även en möjlighet att tillfoga icke-geometrisk information som ex vis toleranser och ytjämnhets-

värden vilka lagras tillsammans med segmentet som properties.

Denna teknik kommer vi att använda oss av för att ge en så fullständig produktmodell som möjligt. Bland annat vill vi testa möjligheten att knyta ett antal verktyg till varje segmenttyp. Verktygen skall rangordnas efter hur pass väl lämpade de är för bearbetning av segmentet med hänsyn till aktuella parametrar. Den lagrade informationen kan användas i en rutin för automatiskt val av verktyg enligt konceptet som finns beskrivet i STU-rapporten 74-4121.

Varje fullbordad geometridefinition avslutas i en rutin som dels frågar programmeraren om det genererade segmentet stämmer överens med ritningen och dels frågar till vilken nivå i systemet programåterhopp skall ske. Den geometriska definitionen av en detaljs geometri består således av att systemet leder programmeraren genom dialogsteg 6 lika många gånger som det finns segment i konturen. När en kontur är klar sker återhopp till närmast ovanför liggande dialogsteg. Här bestäms en ny konturtyp och systemet byter till annat layer samt går vidare till geometridefinitionen enligt dialogsteg 6. Om alla konturerna är färdiggenererade sker istället återhopp till närmast ovanför liggande dialogsteg. På detta sätt arbetar programmeraren sig igenom sin uppgift och återkommer när arbetet är klart till funktionen systemstart vilket innebär att den genererade geometrin finns lagrad med angivna beteckningar. Programmering av en ny detalj kan nu börja.

6 PROGRAMTEKNISKT SYSTEMUTFÖRANDE

Det programtekniska genomförandet av IPS-systemet består av två principiellt skilda etapper.

I det första skedet avsåg vi endast att demonstrera arbetsmetodiken med geometrisegment och interaktiv grafisk teknik. För att visa detta användes ett parameterorienterat språk PEP, som ingår i computervisionsystemets grundprogramvara. I en andra etapp har segmenten byggts upp av fortranrutiner som länkats ihop och bildar tillämpningsprogram som kan samköras med computervisions ordinarie programvara.

Detta kapitel beskriver förutsättningar, villkor och motiv för användande av PEP respektive fortranversionen av systemet.

6.1 PEP-VERSION

I Computervisionsystemet ingår en s k PEP-modul som närmast kan beskrivas som ett antal parameterstyrda geometridefinitionsinstruktioner. Parameteruppsättningarna är relativt lika de som används för geometridefinitionen i APT. I PEP-modulen ingår vidare de vanliga rutinerna för att konstruera programslingor, matematiska funktioner och logiska funktioner.

Eftersom programmering med PEP är ett snabbt sätt att arbeta med Computervisionsystemet, och vi ville åstadkomma ett program för att åskadliggöra det i STU-rapporten föreslagna arbetssättet med interaktiv grafisk teknik, genomfördes en demonstrationsversion av systemet. (Eriksson 77-02-09.)

Det huvudsakliga målet med demonstrationsversionen var att visa att det inte är någon utopi att konstruera ett system som bygger på en dialog där instruktionsinmatningen sker genom menyteknik och där tangentbordet endast används för att ange parametrarnas numeriska värden. Som ett bevis på möjligheten programmerades den i fig 13 visade trycklageraxeln med hjälp av PEP-versionen av systemet.

Första steget i arbetsgången är att man väljer lämplig segmenttyp genom att indikera med ljuspennan på en meny som visas på bildskärmen. Systemet konstaterar vilket segment som valts och visar en ny bild, som detaljbeskriver vilka parametrar som kan förekomma i samband med valt segment. Med hjälp av ljuspennan kan programmeraren ange vilka parametervärden som finns med på ritningen och därefter ange dess numeriska värden med hjälp av tangentbordet. På så vis fyller programmeraren i en systemintern tabell över parametrar som entydigt definierar segmentet. Så snart systemet fått tillräcklig information genereras och visas segmentet på bildskärmen automatiskt. Genom att tolka ritningen och mata in parametervärden till systemet för segment efter segment, genereras den geometriska beskrivningen av detaljen.

Man kan tycka att snabbaste sättet att komma vidare till produktivt utnyttjande av metoden skulle vara en utvidgning av antalet segment till det antal som finns med i det fortranbaserade

systemet. Tyvärr har PEP-systemet några allvarliga begränsningar som inte kan accepteras i ett system som skall användas för produktionsändamål.

Den främsta begränsningen hos PEP-systemet är att man inte kan ange en parameter relativt ett tidigare definierat segment genom identifiering med ljuspenna. Denna definitionsmetod är en av de stora fördelarna med interaktiva grafiska system och som normalt används mycket flitigt vid datorstödd ritningsframställning. Betydelsen av denna begränsning framgår när man betraktar hur de i figur 12 beskrivna modifierande segmenten definieras. Ett modifierande segment definieras till sin form av direkta parametrar men placeras in på den grova geometribeskrivningen på bildskärmen genom att man identifierar de omgivande segmenten med ljuspennan.

Den andra allvarliga begränsningen med PEP-versionen är att man inte kan variera graden av datorstöd efter den enskilde programmerarens behov. För vissa programmerare som arbetat länge med systemet kan det upplevas som tåligt att ständigt behöva se och arbeta med hjälpbilderna. Under utvecklingsarbetets gång har vi sett att ett alltför instruktivt system upplevs som mycket trögt och därför har vi istället satsat på en dialog med varierbart datorstöd.

Den tredje begränsningen i PEP-versionen stör kanske inte mycket i inledningsskedet men är desto mera besvärande för de automatiseringshöjande etapperna av IPS-utvecklingen. I PEP-systemet kan man nämligen inte utnyttja de rutiner som hanterar icke grafisk information. Man går då miste om möjligheten att tillfoga bearbetningsinformation till segmenten vilket är den stora utvecklingspotentialen hos ett system som IPS. Om denna möjlighet bortfaller kan man aldrig åstadkomma det datorstöd för snittuppdeling, verktygsval och skärdata som skisserats i STU-rapporten.

Vi kunde således på goda grunder fatta ett beslut om att övergå till fortranprogrammering och en sammankoppling med det ordinarie computervisionsystemet.

6.2 FORTRANVERSION

Eftersom en total beskrivning av Computervisionsystemets uppbyggnad skulle föra för långt in på detaljer, kommer endast en förståelsen av IPS-programmen nödvändig beskrivning att ges. För en noggrannare beskrivning hänvisas till dels min kollega Dan Johannessons arbete och dels Computervisions systemdokumentation,

Computervisionsystemet är uppbyggt av ett mycket stort antal fortranrutiner för datahantering, terminalhantering, input-output, grafisk behandling etc. Dessa s k systemrutiner arbetar på mikro-nivå och utför endast en mycket begränsad operation på dataflödet. Genom att länka ihop dessa fortranrutiner till större paket s k coreloads åstadkommer man programavsnitt för olika ändamål som t ex beslutslogik, definitionsprogram, generering av matematisk representation eller display av geometrisk information. Även coreloads finns det ett stort antal av, något hundratal, och genom

att kalla in dessa i en bestämd följd i minidatorns primärminne skapas olika instruktioner för datorn. En instruktion kan exempelvis vara interaktiv generering av en linje.

Computervisionsystemet kan beskrivas av fig 14 där det framgår att den centrala tolkningen av allt som matas in i datorn sker i en s k verb-substantiv processor som dels konstaterar vilken handling som skall utföras (verb) och dels konstaterar på vilket data-baselement handlingen skall utföras (substantiv). Denna verb-substantivprocessor är ingångsporten för alla kommandon man vill ge till systemet och hit återvänder programmeraren alltid mellan instruktionerna. Det förekommer heller aldrig några kopplingar mellan två instruktioner som utgått från verb-substantivprocessorn utan varje utförd instruktion är helt oberoende.

Efter verb-substantivprocessorns tolkning av en instruktion kallas ett coreload in som medger val av begränsningar eller modifieringar av den givna instruktionen. Det kan t ex röra sig om att linjen som skall genereras har bivillkoret att den skall ansluta tangentiellt till två cirklar. Bivillkoret matas in från terminalen och tolkas i modifierings-processorn (se fig 14). När detta är klart matas eventuella numeriska värden in till systemet och det avsedda elementet genereras och skrivs i databasen. Från och med att elementet har definierats brukar det visas på bildskärmen och därefter återgår systemet till verb-substantivprocessorn för att taga emot nya instruktioner.

Genom att instruktionerna inte kan kopplas ihop i tillämpningsbundna följder och instruktionsrepertoaren är omfattande, kan man inte begära att andra än datatekniskt erfaren personal skall kunna utnyttja systemet rationellt. Programmeraren kommer alltid att behöva fundera över hur instruktionerna skall följa på varandra och hur han skall planera och genomföra ett visst arbetsmoment. Det blir således alltid programmeraren som måste svara för uppläggnings- och initiering av de enskilda instruktionerna.

Med IPS-rutinerna vill vi ändra på detta krav och har därför byggt upp en tillämpningsdialog, som lotsar programmeraren igenom instruktionssekvensen som behövs för att åstadkomma den avsedda verksamheten.

IPS kan ses som en instruktion inom CADDS. Denna instruktion innehåller emellertid ett mycket stort antal möjliga aktiviteter (se kapitel 5.3).

Genom att skapa en tillämpningsbunden dialog där systemet frågar efter uppgifter och ger programmeraren menyer över möjliga svar har vi eliminerat kravet på en övergripande programuppläggning av programmeraren. Systemdialogen har tagit över den många gånger abstrakta programmeringsmetodiken och ger programmeraren antingen stödinformation genom förklarande bilder på bildskärmen eller genom uppräknings av ett antal alternativa beslutsmöjligheter.

Det är följande två viktiga faktorer som gör att vi tror att en mindre programmeringstekniskt kunnig person skall kunna använda IPS för att generera geometriinformation för svarvning.

- Systemdialogen som leder programmeraren.
- Stödinformationen som hjälper programmeraren.

De olika programmen som bildar IPS är skrivna i fortran och ihop-länkade till coreloads på precis samma sätt som Computervisions standardprogramvara. IPS-programmen består av en metodorienterad systemdialog, menyhantering, grafisk stödbildshantering, definitionsrutiner, genereringsrutiner, rutiner för hantering av till-läggsinformation och rutiner för databashantering. De nyprogram-merade avsnitten är främst inom dialog-, meny-, stödbilds- och definitionstekniken medan Computervisionsystemets genererings-rutiner och databashantering av geometrielement kunnat användas efter smärre modifieringar.

En översikt över hur dialogen för geometribeskrivning av svarv-programmering är uppbyggd, ges av blockschemat i fig 15 , där beteckningar och omfattning av använda egna coreload anges längst till höger. Från dessa egna coreload göres hopp till Computer-visionsystemets olika delar för nämnda generering av geometri-element och databashantering. Genom att återhoppen sker till systemdialogen i stället för till verb-substantiv processorn har en kontinuerlig dialog åstadkommits.

Vidare har IPS-rutinerna gjorts som fristående tillägg till Com-putervisionsystemet så att man slipper alltför omfattande ändrings-arbete vid systemuppdateringarna. Eftersom Computervision genom-för ett antal systemrevisioner per år är det betydelsefullt att de ändringar som då införes inte kan påverka IPS-rutinerna.

6.3 GRAFISK BILDHANTERING

För att kunna visa stödinformation i form av grafiska bilder på ett snabbt och för programmeraren enkelt sätt har en speciell bildhanteringsrutin byggts upp. Rutinen utnyttjar Computervision-systemets layer-teknik, zoomningsteknik och möjligheten att läsa in och överlagra på skivstationen befintliga figurer till den aktuella. För att få en så användbar rutin som möjligt är den helt fristående och kan tillfogas och anropas var som helst i ett befintligt program.

Bildväxlingarna bygger på att Computervisionsystemet kan byta bildutsnitt mycket snabbt genom kommandot ZOOM WIN om den önskade bildens diagonala ändpunkter är kända. Eftersom allt program-meringsarbete utföres i 1:a kvadranten (jfr kap 5.3) bestämdes det att all stödbildsinformation skulle placeras i tredje kvad-ranten, tillräckligt långt bort från origo för att inte ge över-lappning vid bildväxlingar.

Stödbildernas diagonalkoordinater D1 och D2 (jfr fig 16) är konstanta och definierade i samband med att de ritades och lag-rades på skivstationen. Diagonalkoordinaterna för den aktuella detaljens bildutsnitt C1 och C2 är däremot direkt beroende av detaljens dimension. Beräkningen av C1 C2 (jfr fig 16) sker i samband med skalfaktorberäkningen. Bildskärmens bredd : höjd-proportioner där den konstruerade detaljen skall placeras in är 10:8. För att få fram en lämplig skala konverteras de i dialog-

steg 4 givna måttuppgifterna om längd och maximal diameter som gav punkten A, till en rektangel i proportionerna 10:8 med diagonalens ändpunkter i origo och punkten B konstruerad så att punkten A innehålles.

En zoomning direkt över denna rektangel skulle ge svårigheter vid digitaliseringar längs bildens ytterkanter och därför utökas diagonalen till C1 och C2. Med punkterna C1, C2 och D1, D2 har vi två bildutsnitt som inte överlappar varandra och mellan vilka systemet kan växla inom någon sekund.

I båda bildutsnitten använder vi layerteknik. För den aktuella programmeringsuppgiften är det konturtypen som bestämmer vilket layer som skall vara aktivt och synligt. Stödbilderna är uppdelade i layer enligt kap 5.3 och där finns stort utrymme för ytterligare bildinformation.

De funktioner som utföres av bildhanteringsrutinen är först en avkodning av vilket layer som skall vara aktivt och vilka som skall vara synliga och därefter en avkodning av vilket bildutsnitt som skall visas. Genom att arbeta på detta vis kan nära nog oändlig mängd bildstödsinformation byggas in i systemet efterhand som det anses önskvärt.

7 SYSTEMREALISERING OCH VIDAREUTVECKLING

I framtiden kommer man i stor utsträckning att ersätta manuella individberoende metoder med datorstödda rutiner där man lägger stor vikt vid att den erforderliga informationen skall kunna utnyttjas rationellt. Detta indikerar att framtida system för automatisk beredning av NC-maskiner kommer att arbeta dialogt och med omfattande datorstöd. Dessutom måste systemen med relativt enkla medel kunna modifieras när nya tillverkningsmetoder införes.

För att åstadkomma den förenklade metodorienterade programmeringsteknik, som krävs om man vill återföra programmeringsarbetet till maskinoperatörerna, kommer de interaktiva grafiska systemen att spela en mycket stor roll.

Den vidare utvecklingen av IPS-systemet sedan de nu utvecklade rutinerna för generell geometribeskrivning testats under produktionsmässiga förhållanden, bör inriktas på att komplettera systemet med s k produktspecifika segment och att börja utnyttja den icke geometriska informationsbehandlingen för hantering av kataloginformation för verktygs- och skärdataval.

Det första vidareutvecklingssteget är därför att man tager systemet i drift för att genom praktiska prov utröna vilka produktspecifika segment som med nödvändighet måste tillfogas systemet. Genom de inledande undersökningarna som utfördes i maskinverkstaden kunde man konstatera att även till synes komplicerad geometri kunde behandlas med de generella segmenten. I många fall där man i verkstaden har bedömt en detaljs geometri som svårdefinierad har segmentmetoden visat sig vara en enklare beskrivningsmetod än tillgängliga alternativa metoder.

Ett huvudskäl till att beskrivningen med segment oftast blir enkel är dess uppdelning av beskrivningsarbetet i tre faser. Den första fasen där detaljens grova drag beskrivs med huvudsegment förorsakar sällan några besvär eftersom cylindriska, koniska och planbearbetade ytor oftast är fullständigt mått-satta på ritningsunderlaget. Dessa segment definieras därför mycket snabbt. Den andra fasen av beskrivningsarbetet har då som referens- och utgångsgeometri den grova formen från föregående beskrivningsfas, och det är därför tillräckligt att man storleksbestämmer lämpligt modifieringssegment och anger mellan vilka huvudsegment modifieringen skall införas. Den interaktiva grafiska tekniken möjliggör denna identifiering mycket rationellt genom indikering av respektive segment på bildskärmen. I datorsystemet finns därefter rutiner som beräknar skärningspunkterna mellan de två ursprungliga segmenten och det nytil-lförda modifieringssegmentet. Skärningspunkterna utnyttjas sedan för att skapa en kontinuerlig kontur genom avkortning av huvudsegmenten. Som ett sista moment i geometribeskrivningen kommer den på detta vis genererade konturen att tillfogas de geometriska former som är så produktorienterade att de inte kan beskrivas med hjälp av huvudsegment och modifierande segment. För att inte systemet skall svälla ut i ett stort antal produkt-specifika segment under införandefasen måste något av följande

två kriterier vara uppfyllt innan ett nytt segment skapas och tillfogas systemet.

- Den aktuella geometrin skall bearbetas med specialverktyg eller speciella metoder.
- Den aktuella geometrin är ofta återkommande för produkttypen och går inte att beskriva utan omfattande koordinatuträkningar.

Om inte dessa regler följs vid införandet av nya segment kommer systemet att successivt övergå i ett svåröverskådligt makrobibliotek.

Under den här nämnda perioden för idrifttagande och komplettering av segmenttyperna bör systemmodellen som beskrevs i kap 5.1.2 t o m 5.1.4 genomföras. Samtidigt med vidareutvecklingen av systemet som skall möjliggöra en förenklad nyprogrammering kommer man då att lösa flera av dagens omedelbara problem kring hanteringen av de befintliga hålremorna.

Med tillgång till en lokalt placerad dator i verkstaden kommer man att kunna förenkla hålremsadministrationen avsevärt jämfört med idag. En av de första åtgärderna som bör vidtagas är därför att koppla ihop minidatorn med verktygsmaskinstyrningarna. På så sätt kommer man att kunna begära hålremsinformationen direkt överförd från datautrustningens skivminne till verktygsmaskinerna. Ett sådant arrangemang kommer dels att minska risken för felaktig inläsning av hålremsinformation och dels minska risken för förväxlingar av program och programversioner. Förutom överföring av hålremsinformationen kommer programbiblioteket att bli mer överskådligt med rutiner för att snabbt taga fram rätt hålremsinformation.

Den verkstadsplacerade utrustningen kommer vidare att medföra förbättrade rutiner för ändring av befintliga program. Genom att införa möjligheten till snabb och enkel programrättning kommer programmerarna och verktygsmaskinoperatörerna att lära sig använda systemet och kunna ge ytterligare synpunkter på vilka krav man måste ställa på ett datorstött beredningssystem.

Under införandefasen kommer på detta sätt personalen i verkstaden att få ökade erfarenheter av arbete med datorbaserade metoder. Denna kunskapsuppbyggnad och vana vid datorhjälpmedel kommer att få mycket stor betydelse för i vilken omfattning verkstadspersonalen kommer att kunna utnyttja framtida produktionssystem. Genom att maskinoperatörerna i en framtida verkstad kommer att frikopplas från den direkta bearbetningsoperationen i numeriskt styrda verktygsmaskiner, kommer deras arbetskapacitet att kunna utnyttjas för andra ändamål. Den stegvis införda och vidareutvecklade programmeringsmetoden kan då utgöra en meningsfull sysselsättning för maskinoperatörerna.

SUMMARY IN ENGLISH

INTERACTIVE GENERATION OF GEOMETRY FOR NC-LATHES WITH GRAPHICAL AND DIALOGUE COMPUTER AID.

This report describes a new approach to the problems involved with part programming of numerically controlled machine tools. The suggested system is specially designed to accomodate customer oriented production and production in small unrepeatd lots. To meet the demands derived from this type of production in medium size factories the system is based on the use of interactive graphics which gives the possibility to create a flexible communication between man and computer.

The system comprises methods for interactive generation of geometry developed by S Å Eriksson dealt with in this report and methods for generation of tool paths developed by D Johannesson treated in a report titled "Generering av verktygsbanor för NC-svarvar på ett interaktivt grafiskt minidator-system" - Interactive generation of tool paths for NC-lathes on an interactive graphics minicomputer system.

Since interactive graphics became available on minicomputers in the early seventies it has proven to be a most useful tool to facilitate for non-specialists to communicate with computers and increase the use of computer aided methods. The investigations carried out in the machine shop at Kockums Shipyard clarified the need for more efficient methods to do the part programming as the number of numerically controlled machine tools increases. Furthermore the undesired withdrawal of the possibilities for machine tool operators to make the technological decisions, that is the result of the introduction of numerically controlled machine tools, can be attacked with the use of operation oriented methods for part programming.

After thorough investigations of the elements influencing the cost of numerically controlled production, the most important demands for profitable production were stressed and considered when the new programming system was specified.

The part programming methods proposed consists of a simplified method for describing part geometry with standard elements on an interactive graphics minicomputer. The interaction between man and computer is aided by a dialogue which is oriented towards the terminology used for the actual operation in the work shop. As the system is meant to be usable for different categories of operators, there is a possibility to choose between a dialogue with little computer assistance and a more extensive with menues and graphical computer aid.

With this approach there will be a fair chance to provide an usable part programming system for the personel in the work shop. The proposed system is based on an interactive graphics system which also can be utilized for datatransmission of control information between a program library stored with the system and the machine tools. Finally the interactive graphics system provides facilities for editing control information.

The extensions made to the interactive graphics system will make it useful for part programming in an environment where the demands are similar to those of the machine shop at Kockums. The implementation of a system of this kind in a work shop can be expected to give economical benefit as well as give conditions for job enrichment by the increased use of computer aided methods.

FIGURFÖRTECKNING

Fig 1	Maskinverkstaden
Fig 2	Kronologiskt schema
Fig 3	Placering av standarddetaljer enl VIS-norm
Fig 4	Kostnader för ianspråktagen produktionsresurs
Fig 5	Tillverkningsspektrum för supportsvarv
Fig 6	Tillverkningsspektrum för karusellsvarv
Fig 7	Systemkonfiguration vid LTH
Fig 8	Informationsflödet vid programmering av NC-maskiner
Fig 9	Sökbegrepp och lagringsformat vid programmering av NC-maskiner
Fig 10	Systemkonfiguration för IPS-system i produktion
Fig 11	Integrationsnivån hos IPS-systemet
Fig 12	Huvudsegment och modifierande segment i IPS-systemet
Fig 13	Trycklageraxel
Fig 14	Kommandostruktur i CADDS
Fig 15	Omfattningen för IPS-coreload
Fig 16	Bildskärmsuppdelning i IPS-systemet



Fig 1 Maskinverkstaden

Kronologiskt schema över forskningsarbetet för att taga fram geometridelen av IPS

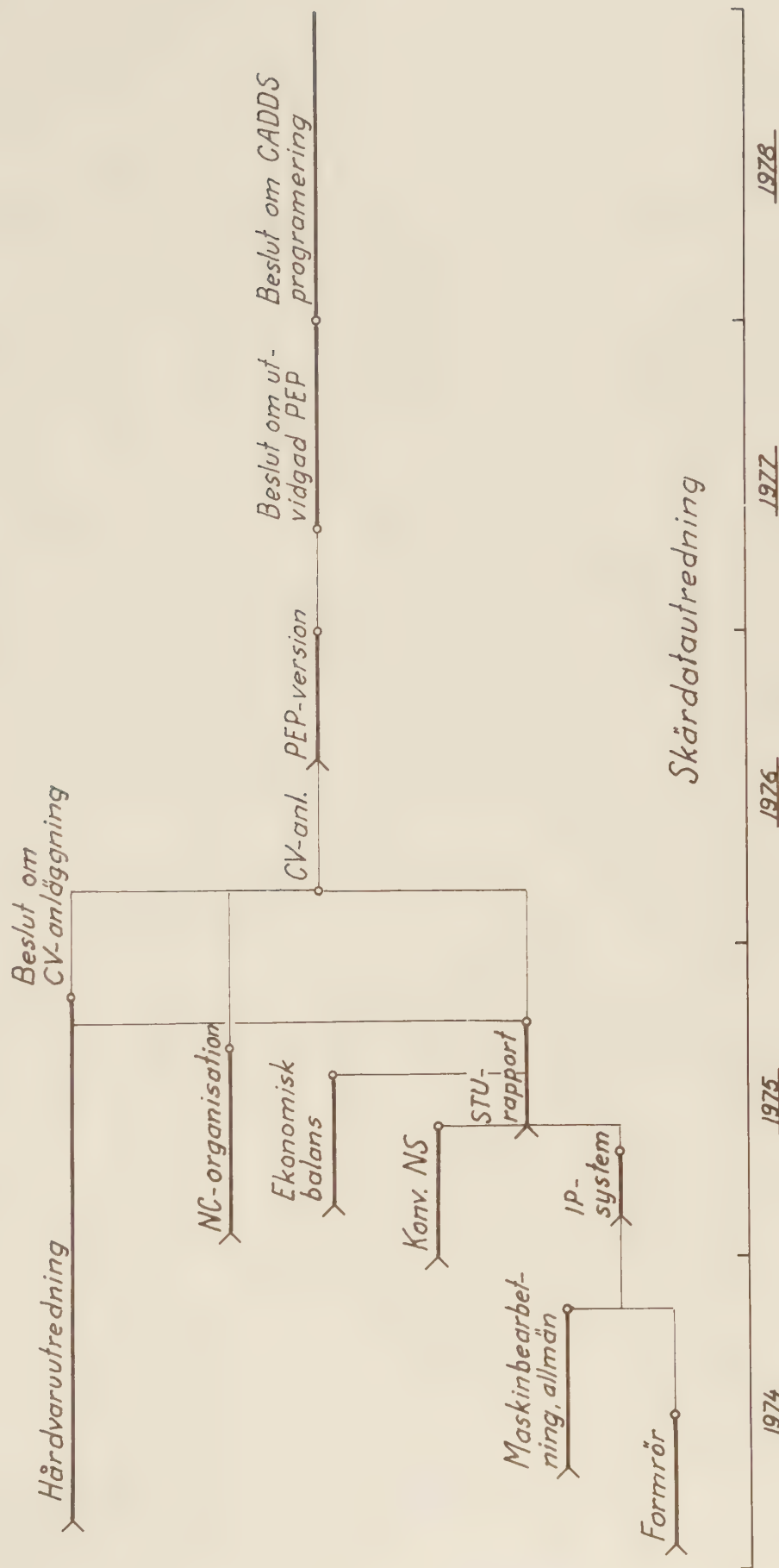
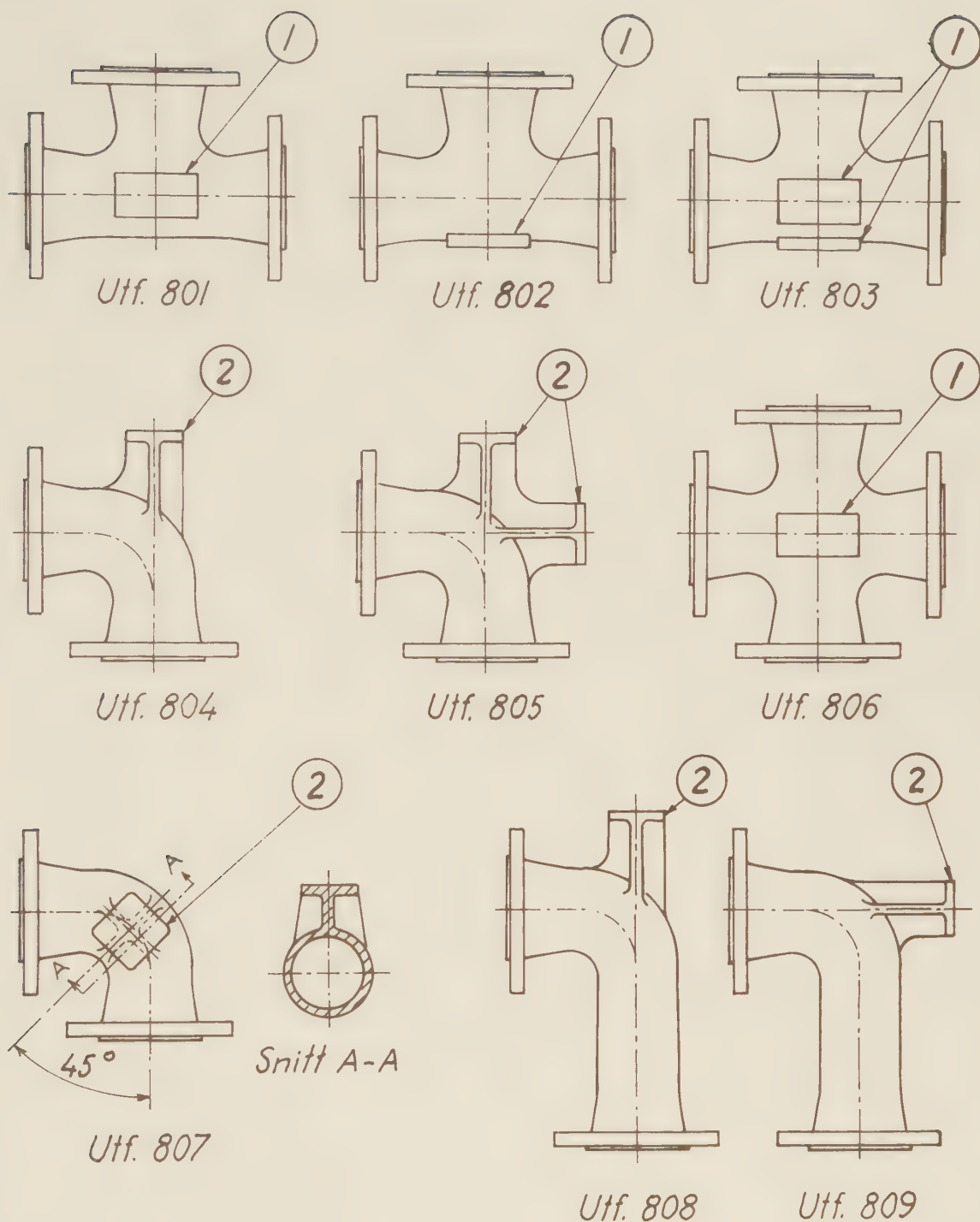


Fig 2 Kronologiskt schema



Stödfot VIS 323 det nr 1
 Stödfot VIS 322 det nr 2

Fig 3 Placering av standarddetaljer enl VIS-norm.

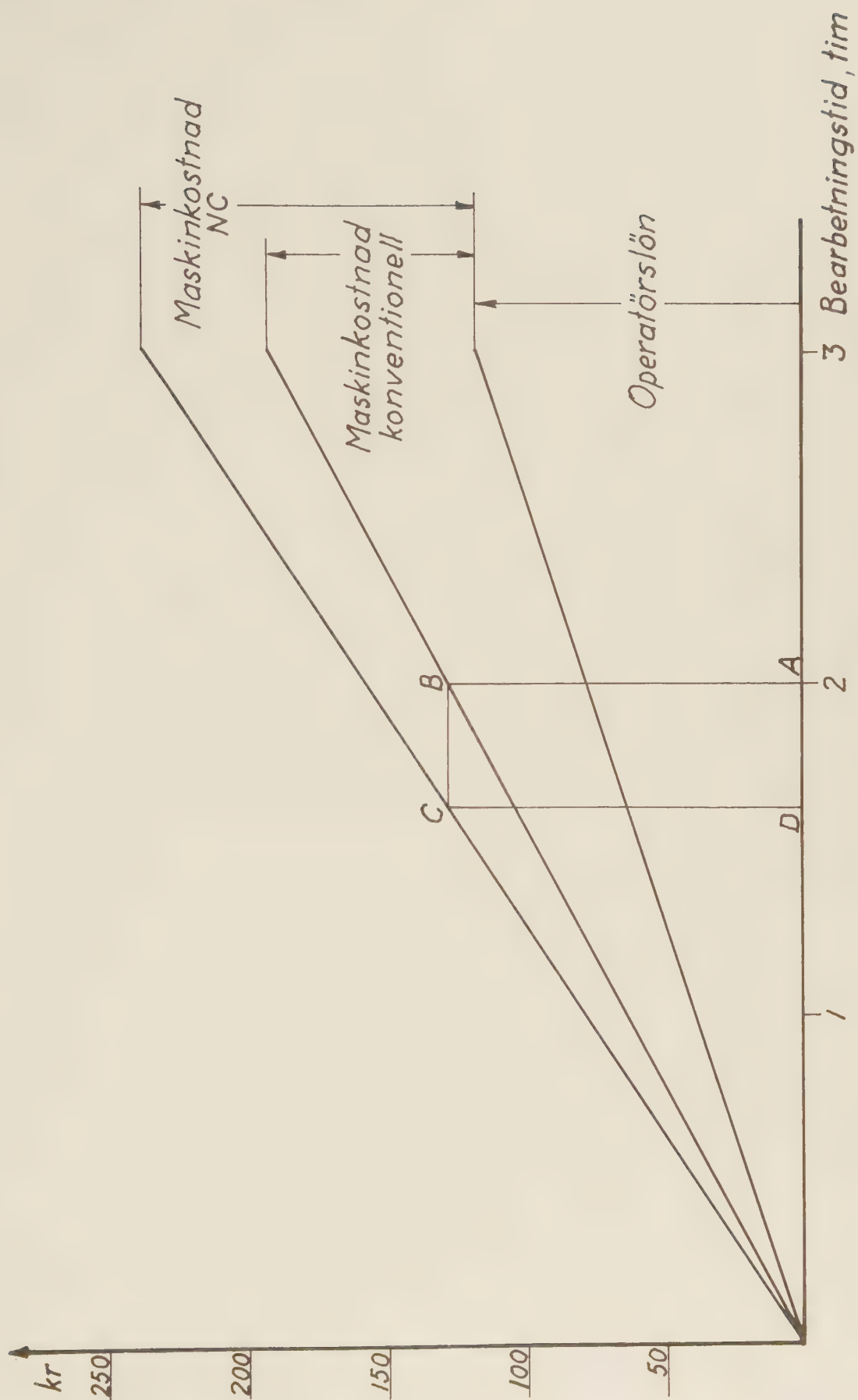


Fig 4 Kostnader för ianspråktagen produktionsresurs.

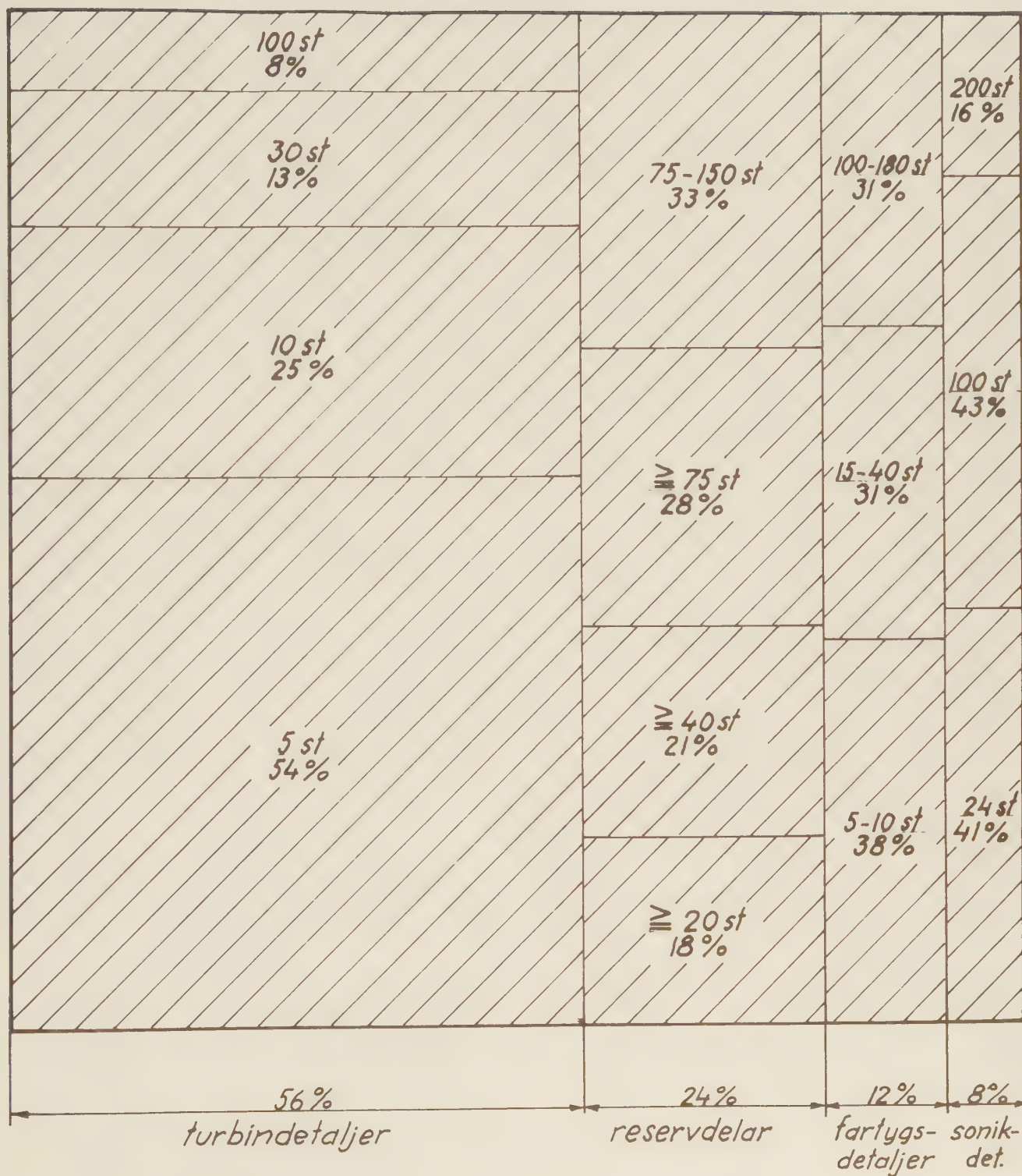


Fig 5 Tillverkningsspektrum för supportsvarv.

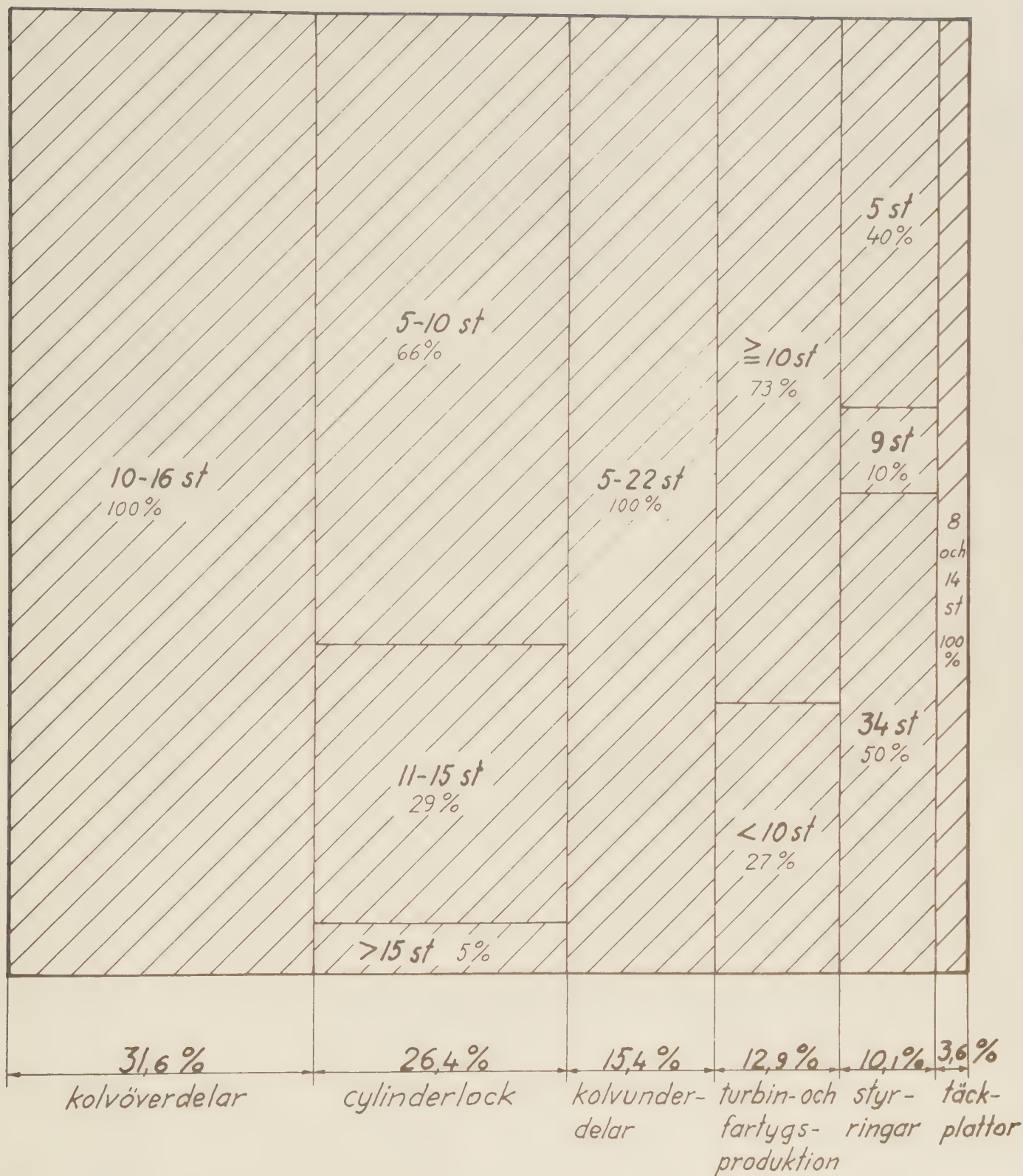


Fig 6 Tillverkningspektrum för karusellsvarv.

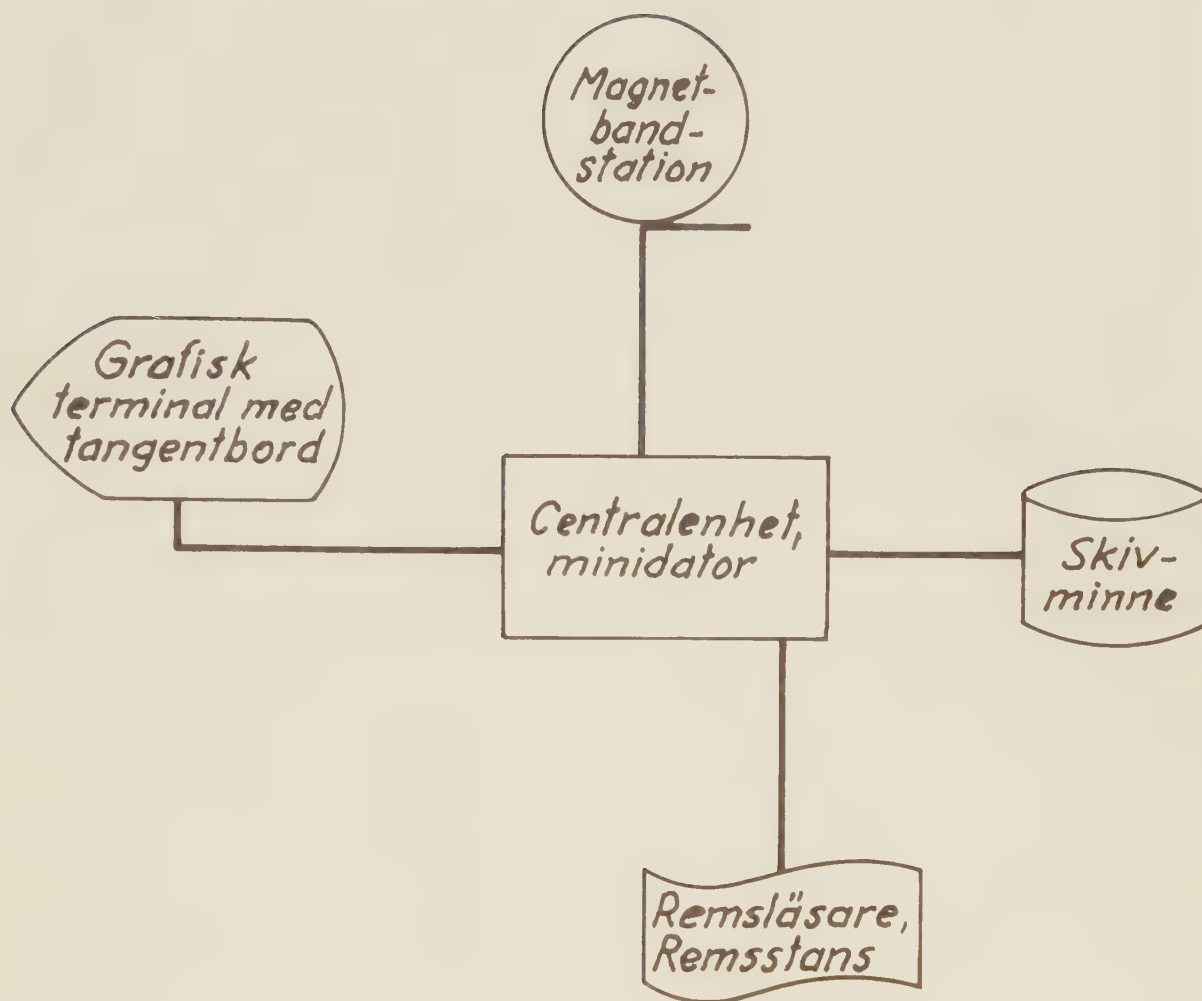


Fig 7 Systemkonfiguration vid LTH.

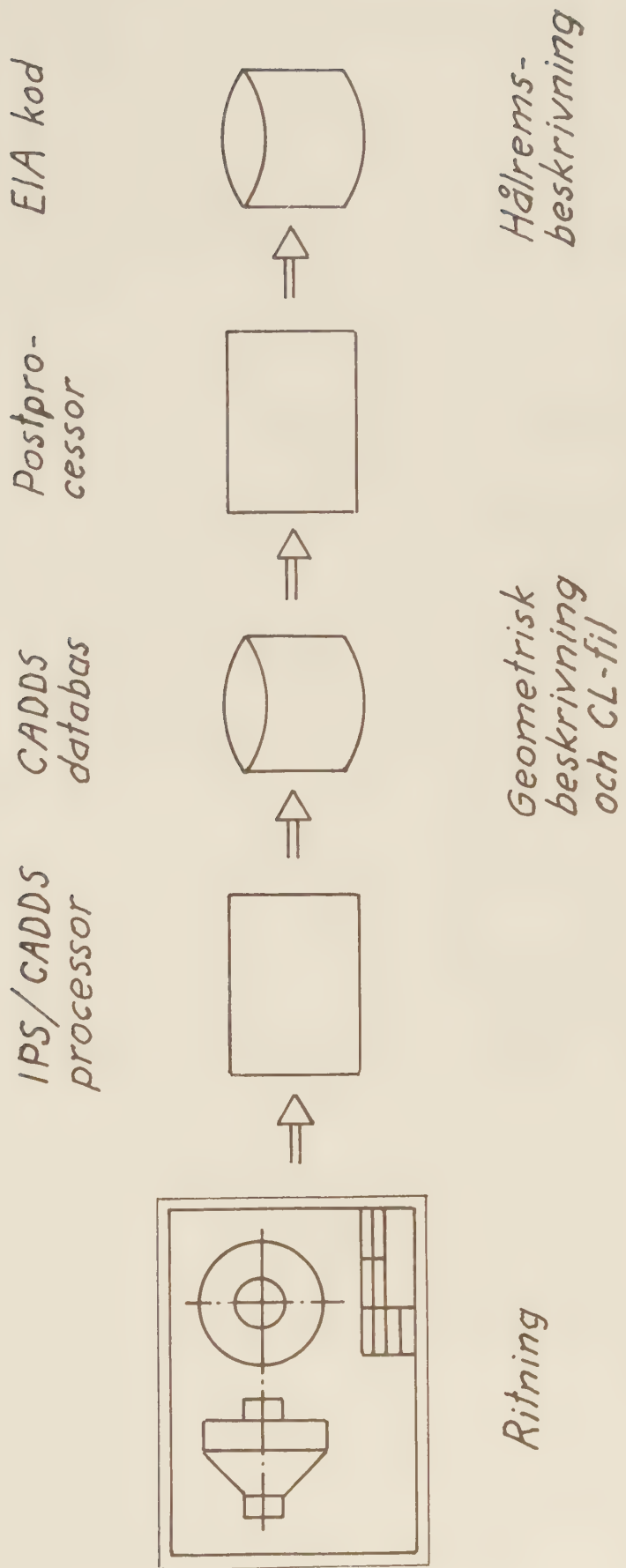


Fig 8 Informationsflödet vid programmering av NC-maskiner.

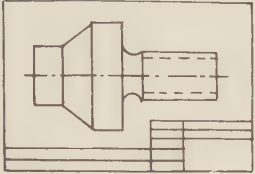
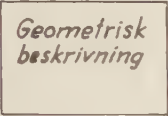
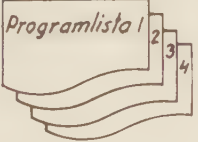
Informationstyp		Sök- och biblioteks- begrepp	Lagringsformat
  <i>Avkodning</i>	<i>Ritningsunderlag</i>	<i>Ritningsnummer</i>	
  <i>Snittuppdelning</i> <i>Bearbetningsdata</i>	<i>Råkontur</i> <i>Finkontur</i>	<i>Ritningsnummer</i> <i>Benämning</i> <i>Programnummer</i>	<i>CADDs databas</i> <i>Geometrielement</i> <i>med tilläggs-</i> <i>information</i>
  <i>Postprocessor-</i> <i>behandling</i>	<i>Uppspänningar</i> <i>Deloperationer</i> <i>inkl. bearbetnings-</i> <i>teknologi</i>	<i>Programnummer</i>	<i>CADDs databas</i> <i>CL- data</i>
 	<i>Hållemsin-</i> <i>formation</i> <i>Hållemsa</i>	<i>Programnummer</i> <i>Programnummer</i>	<i>Textfil</i> <i>EIA-kod</i>

Fig 9 Sökbegrepp och lagringsformat vid programmering av NC-maskiner.



Fig 10 Systemkonfiguration för IPS-system i produktion.

Dialogsteg Icke-geometrisk information

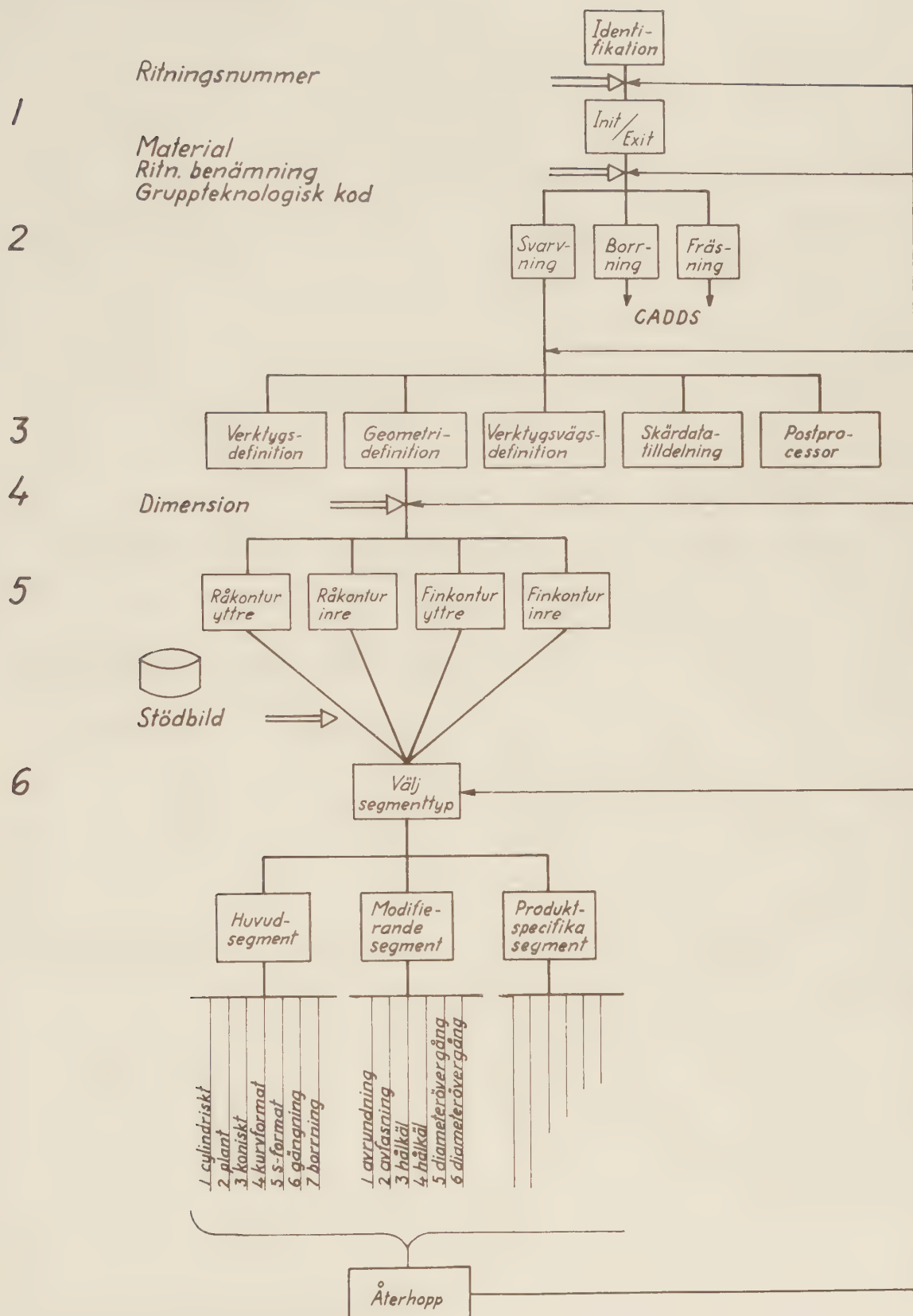
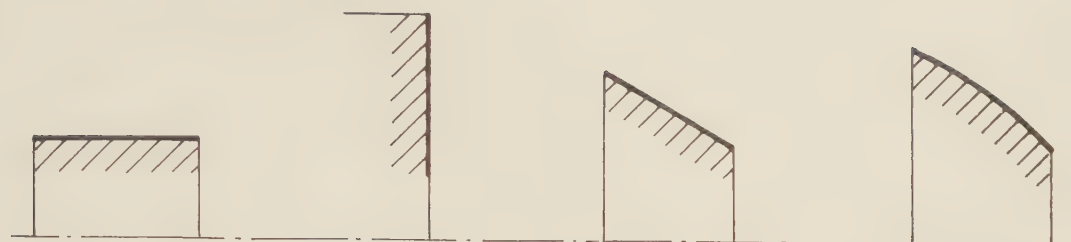


Fig 11 Integrationsnivån hos IPS-systemet.

Huvudsegment



cylindriskt

plant

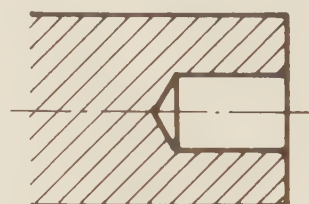
koniskt

kurvformat



s-format

gängning



borrning

Modifierande segment



avrundning

avfasning

hålkäl

hålkäl



diameterövergång



diameterövergång

Fig 12 Huvudsegment och Modifierande segment i IPS-systemet.

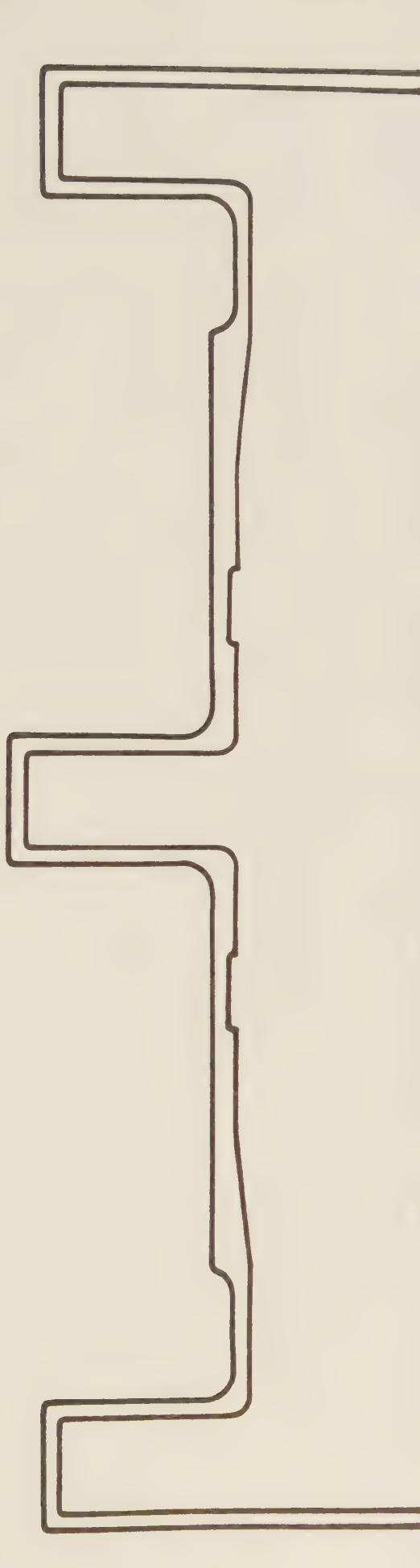


Fig 13 Trycklageraxel.

Språkprocessor		Dataprocessor		Geometriprocessor	
Verb – Substantiv processor	Modifierings-processor	Elementtyp-begränsning	Data-input	Definitions-processor	Databas-hantering
INS LIN	TANTO	CIR	X, Y, Z dig, dig	Grafisk display	Utökad databas
INPUT			OUTPUT		

Fig 14 Kommandostruktur i CADD5.

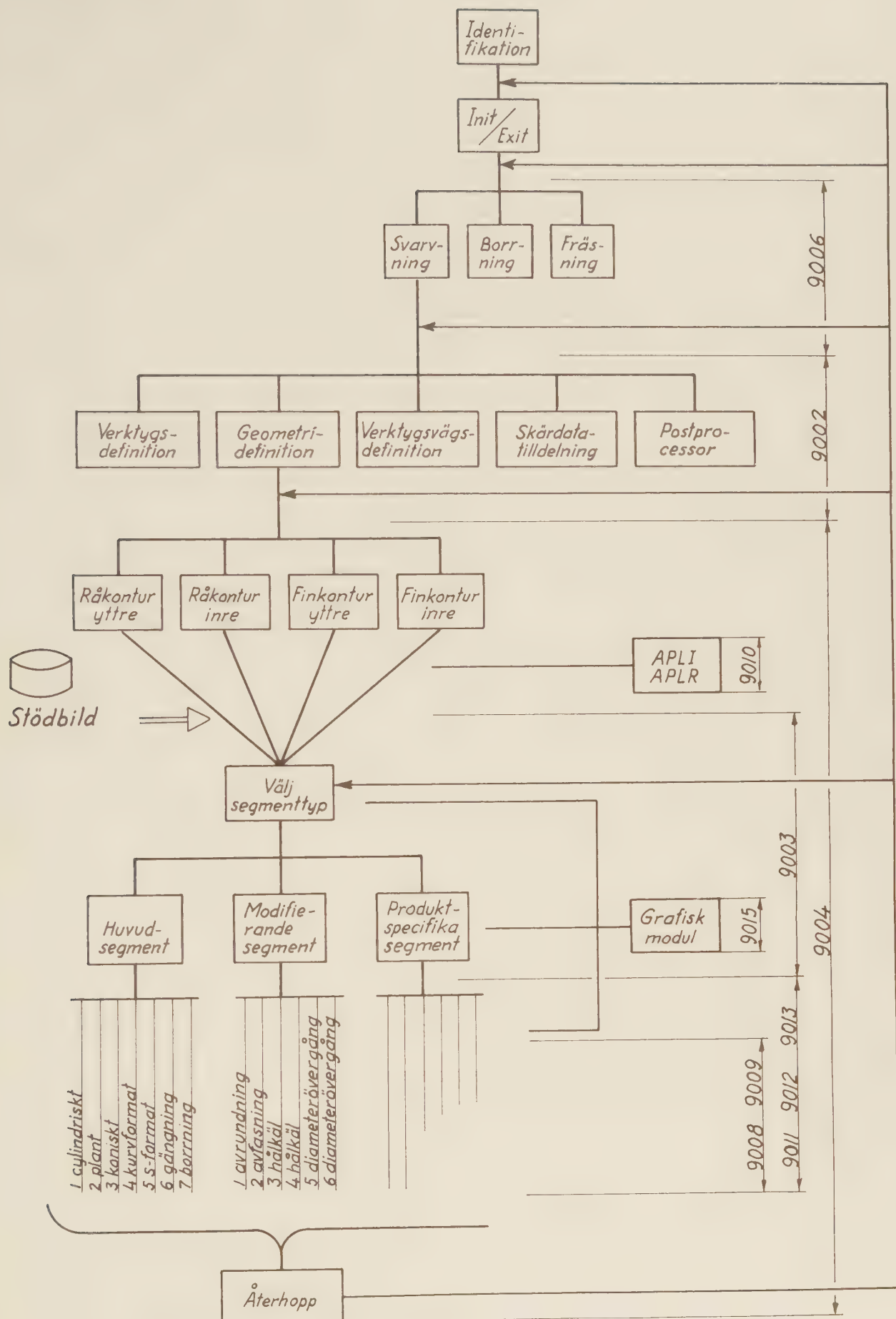


Fig 15. Omfattningen för IPS-coreload.

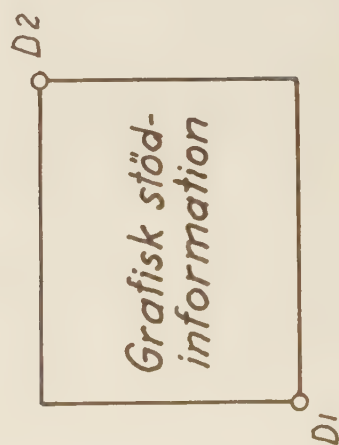
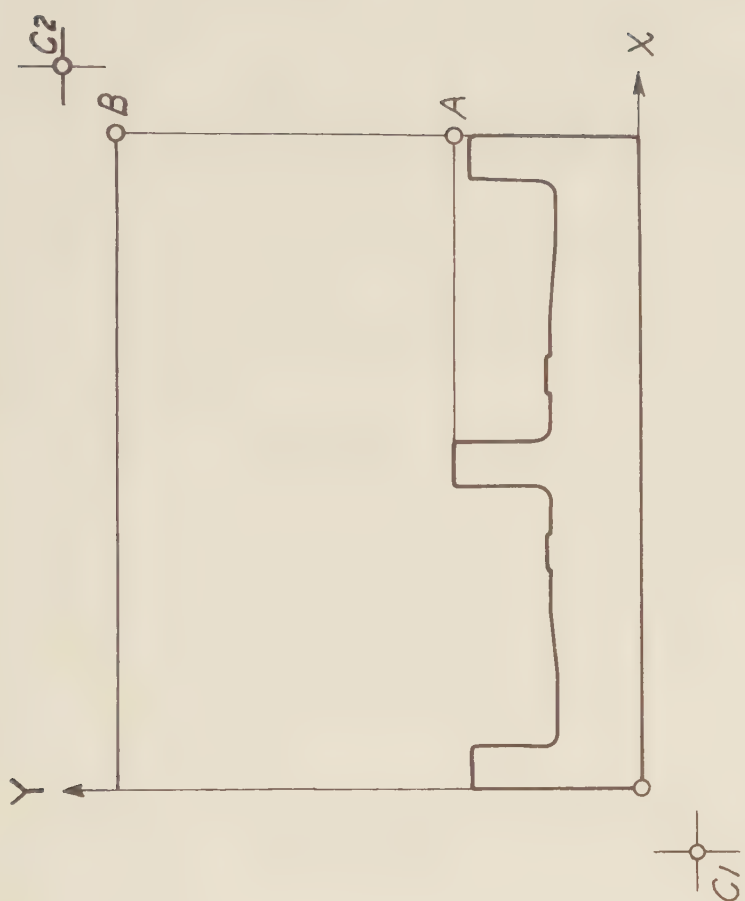


Fig 16 Bildskärmsuppdelning i IPS-systemet.

LITTERATURFÖRTECKNING

Arbnor, I. - Bjerke, B.: Företagsekonomisk metodlära, Studentlitteratur 1977.

Asztély, S.: Investeringsplanering, Esselte Studium 1973.

Bergqvist, J. - Eriksson, S.Å.: Uppföljning av kostnadsförändringar vid övergång från konventionell bearbetning till NS-bearbetning, Intern utredning Kockums 1975.

Björke, Ö.: Produktdefinierande språk, Tapir 1974.

Braid, I.: Designing with volumes, Cambridge 1973.

Dietz 621: Computer System im DNC-Einsatz, Dietz Computer Systeme 1977.

Eriksson, S.Å.: Lägesrapport, PEP, Rapport Kockums 77-02-09.

Eriksson, S.Å.: IPS, Integrerat Produktions System, Intern utredning Kockums 1975.

Eriksson, S.Å.: Metoder att uppnå bättre ekonomisk balans mellan NS-bearbetning och konventionell bearbetning, Intern utredning Kockums 1975.

Göransson, R. - Roth, L.: Förenklingar vid formrörstillverkningen hos Kockums Mekaniska Verkstad AB, LTH 1974.

Hatvany, J.: Computer Languages for Numerical Control, Proceedings Prolamat 73.

Haugerud, B.: Kobra a programming language for description of axially symmetric workpieces, Technical University of Norway 1968.

Hermansson, I.: Verktygs- och skärdataval vid svarvning med NS-maskiner på Kockums Mekaniska Verkstad AB, LTH 1976.

Holm, S. - Svanberg, K.: NC-organisationen, Studie vid Kockums Mekaniska Verkstad AB, LTH 1975.

Holmer, B.: Datorstödd NC-programmering - råd för systemval, IVF-resultat 78649.

Igoshi, M. et al: Development of a graphic display system for turning work (GST), CIRP report 22/1 1973.

Johannesson, D.: Generering av verktygsbanor för NC-svarvar på ett interaktivt grafiskt minidatorsystem, Diss Lund 1979.

Kullberg, G. m fl: Samordnad Datorstödd Produktion, Ingenjörslärläroverket 1976.

Lettström, R. - Månsson, K. - Ralvert, K.: Förberedande ritningsanalys för minidatorbaserad framtagning av ritningsunderlag, LTH 1974.

Lumsden, K.: Teknologisk produktionsstyrning, Diss Lund 1975.

Martin, J.: Design of Man-Computer Dialogues, Prentice Hall 1973.

Nya Fabriker: Gruppering av produktionsutrustning - fem lösningsprinciper, SAF 1978.

Spur, G. - Stute, G. - Weck, M.: Rechnergeführte Fertigung, München Wien 1977.

Spur, G.: Rechnergestützte Zeichnungserstellung und Arbeitsplanung, ZWF, Hanser Verlag 1978.

STU-rapport 74-4121: Integrerad produktionsstyrning för NS-maskiner, Malmö 1974.

Takeyama, H. et al: An Automatic Programming System For Lathe Operations - MELTS, CIRP report 22/4 1973.

